DATAMAN S4

BEDIENUNGSANLEITUNG

Deutsche Übersetzung
des englischem Originals
vom JUNI 1996

Wichtige Notizen

Wenn der S4 kein Lebenszeichen von sich gibt

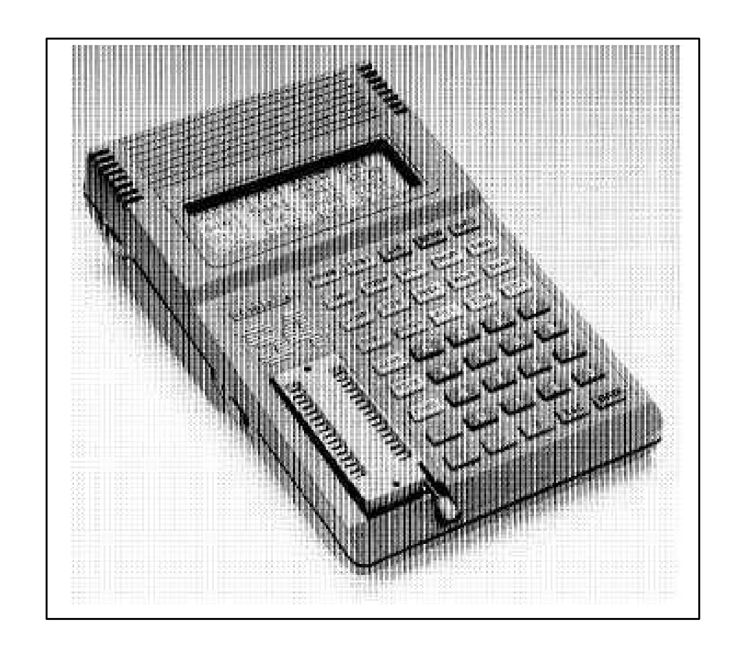
Der **ON/OFF** Schalter ist auf der linken Seite.
Beim Einschalten diesen von Ihnen weg drücken.
Wenn nichts auf dem Display erscheint, das Einstellrad auf der linken Seite drehen, um den Kontrast zu erhöhen.
Wenn Sie noch immer keine Anzeige haben, könnte der S4 komplett entladen sein.
Drücken Sie die **RESET** Taste für ein paar Sekunden.
Die **RESET** Taste befindet sich in einem kleinen Loch unter dem **ON/OFF** Schalter. Wenn Sie nichts finden womit Sie die Taste drücken können, benützen Sie den 2mm Stecker auf dem Write-Draht.
Drücken Sie nicht zu fest. Die Taste ist nämlich ein Schalter mit kurzem Schaltweg.
Der S4 sollte starten, wenn Sie die Taste loslassen.

Halten Sie die Akkus geladen

Wenn der S4 entladen ist und nicht funktioniert, dann müssen die Akkus sofort geladen werden.
Ein guter Grund ist, daß NiCd Zellen ihre Polarität ändern können und so die Lebenszeit des Produktes verkürzt wird.
Ein anderer Grund ist, daß die Lithium Batterie verbraucht wird.
Lassen Sie den S4 niemals ungeladen. Laden Sie ihn so schnell als möglich.

Einführung in S4

Der DATAMAN S4 ist ein batteriebetriebenes **PROM** Programmiergerät.
Er beinhaltet 128k, 256k oder 512k **RAM** , welches Daten speichert und bei jedem Einschalten neu konfiguriert werden kann. In das **RAM** können die Programmdaten geladen werden, welche mit dem Computer, über ein RS232 Interface, oder der S4 Tastatur verändert werden können. Der S4 ist ausgestattet mit einer Plug-In-Emulation für einen 24/28/32 Pin Emulator. Dies hat den Vorteil, daß ein neues Programm mit einer Emulation getestet werden kann. Wenn es funktioniert kann ein **PROM** in das System gesteckt und programmiert werden und die Arbeit ist erledigt.
Das Programm beinhaltet **EPROMS** aus der 27 Serie, wie zum Beispiel 2716 oder 278000, es beinhaltet weiters **FLASH-EPROMS** und die meisten **EEPROMS**, inklusive den 28, 52, 55 & 98 Serien.
Andere Bauteile, wie zum Beispiel Single Chip Mikroprozessoren, benötigen einen entsprechenden Adapter.



Checkliste der Teile und des Zubehörs

1. Dataman S4
2. Handbuch (Sie lesen es gerade)
3. Write Draht 2mm Stecker mit Abgreifklemme
4. **EMULead** - Flachbandkabel mit 32 Pin DIL Stecker
5. **LIBRARY ROM**
6. Diskette mit Treibersoftware und Zubehör
7. Ladegerät

Drei Jahre Garantie

Der S4 hat eine Herstellergarantie, von 3 Jahren , ab Kaufdatum.
Diese betrifft alle elektronischen Teile mit Ausnahme der Akkus und der Lithiumbatterie. Bei dieser beträgt die Garantiezeit 1 Jahr.
Bei einer Reklamation rufen Sie zuerst unsere Serviceabteilung an und geben Sie die Fehler durch. Einige Probleme können von uns sehr schnell gelöst werden, ohne daß wir das Produkt benötigen. Sie können zum Beispiel neue Batterien einsetzen. Falls wir das Produkt doch benötigen, bekommen Sie von uns eine Rücknahme Nummer. Sie sollten daraufhin das Produkt schnellsten zu uns schicken und sich Ihre Rücknahme Nummer merken. Schicken Sie das Gerät direkt zu uns, denn wenn Sie das Gerät zu einem Händler schicken, kostet Sie das nur unnötige Zeit. Wir versuchen das Gerät in weniger als 2 Werktagen zu reparieren. Es hilft uns sehr, wenn Sie eine Beschreibung der Produktfehlern beilegen.

Überprüfung ob der S4 funktioniert

Schalten Sie den S4 ein und drehen Sie das Display Einstellrad von Ihnen weg bis auf sein Maximum. Das Display sollte jetzt dunkler werden. Stellen Sie das Einstellrad so ein , daß der beste Kontrast entsteht. Dieser hängt von der Temperatur und dem Betrachtungswinkel ab. Wenn ein **LIBRARY** Programm geladen ist, dann zeigt der S4 die Versionsnummer an:

```
DATAMAN S4 LIB 1.1
128K BYTES RAM
>
```

Wenn diese Nachricht erscheint, dann ist der S4 betriebsbereit. Lagern Sie das **LIBRARY ROM** auf einem Stück antistatischem Schaumstoff. Wenn nichts auf dem Display erscheint, dann verändern Sie den Kontrast abermals auf das Maximum. Wenn noch immer nichts passiert, überprüfen Sie, ob die Akkus geladen sind. Stecken Sie das Ladegerät an, während Sie die **RESET** Taste für ein paar Sekunden drücken. Wenn Sie loslassen, sollte folgende Nachricht erscheinen:

```
DATAMAN S4 BIOS 1.1
128K BYTES RAM
NO LIBRARY LOADED
```

Laden des LIBRARY ROM

Im Normalfall ist es nicht notwendig das **LIBRARY ROM** zu überschreiben, außer wenn eine neue Version des Arbeitsprogrammes geladen werden soll.
Das laden des **LIBRARY ROM** funktioniert so: Drücken Sie die **RESET** Taste mit dem Write Draht Stecker. Die Taste befindet sich in dem Loch neben dem **ON/OFF** Schalter (Wenden Sie keine Gewalt an, es reicht ein kleiner Druck auf die Taste).
Sie sehen jetzt diese Nachricht:

```
DATAMAN S4 BIOS 1.1
128K BYTES RAM
START LIBRARY?
```

Sie werden merken, daß die **LIBRARY** Versionsnummer nicht identisch ist, weil kein **LIBRARY** Programm geladen ist. Der S4 glaubt eine gültige **LIBRARY** Version vorliegen zu haben und fragt, ob Sie diese starten wollen.

ESC = NO
ENTER = YES

Wenn Sie eine neue **LIBRARY** laden wollen, dann drücken Sie **ESC** und anschl. **LIB**. Stecken Sie das **LIBRARY ROM** in den Sockel und drücken Sie **ENTER**. Der S4 lädt jetzt das im **LIBRARY ROM** enthaltene Programm. Das Programm startet erneut und das Programm wird geladen. Wenn das **LIBRARY** Programm ordnungsgemäß geladen wurde, zeigt der S4 die Versionsnummer am Display. Wenn die **RESET** Taste gedrückt wird, geht der S4 in den Low Level **BIOS MODE** über in dem Sie nur **LIBRARY PROMS** der 27256 Serie programmieren können. Sie sollten den S4 nicht im **BIOS MODE** verwenden, ohne ein **LIBRARY** geladen zu haben.
Theoretisch sollten Sie nie dazu gezwungen sein, das LIBRARY Programm laden zu müssen. Das LIBRARY Programm ist schon in Ihrem S4 gespeichert wenn Sie ihn kaufen. Im S4 ist ein permanenter Speicher eingebaut, in dem Programme und Daten gespeichert sind.

Der einzige Grund, ein **LIBRARY** zu laden, ist, wenn Sie Ihr System mit einer neueren Version aufrüsten wollen, oder wenn Sie eine alte Version laden wollen, nachdem jemand diese verändert hat.

RESET & BIOS Modus

Wenn die **RESET** Taste gerückt wird, lädt der S4 ein Programm, daß im versteckten **ROM** des Mikrocomputers, genannt **BIOS** (**B**asic **I**nput/**O**utput **S**ystem), gespeichert ist. Während der S4 im **BIOS** Modus läuft, funktionieren zwar die meisten Befehle, es wird aber nicht empfohlen, dieses ohne einer im **TPA** (**T**ransient **P**rogram **A**rea) geladenen **LIBRARY** zu machen. Das **BIOS** beinhaltet Subroutinen, die für das Bearbeiten von Ein- und Ausgaben benötigt werden - z.B. RS232, Keyboard, Display etc. Der **BIOS** Modus ist nur für die Entwicklung neuer **LIBRARY** Programme gedacht. Im **BIOS** Modus können mit dem S4 nur 27256 **EPROMs**, wie z.B. das **LIBRARY ROM**, programmiert werden.

Dieser Abschnitt ist nur zur Komplettierung geschrieben, da es unwahrscheinlich ist, daß jemand eine neue Software für den S4 im BIOS Modus, entwickelt.

Wird die **RESET** Taste gedrückt während das Ladegerät angeschlossen ist, bekommen die Akkus einen Stromstoß. Das verursacht einen „ Jump Start“. Dies ist bei Akkus, die absolut entladen sind, notwendig.
Am S4 Mikroprozessor und am **RAM** liegt immer eine Spannung an.
Beim Einschalten wird der **SLEEP MODE** beendet und das System startet bei der Adresse, auf die mit dem **WARM START VECTOR (WSV)** gewiesen wird. Der **WSV** wird verwendet, um das System beim richtigen Anfangspunkt zu starten. Wenn **RESET** gedrückt wird, weist der **WSV** auf das **BIOS**, aber zuerst wird auf das **TPA** geschaut. Wenn dort ein gültiges **LIBRARY** vorhanden ist, werden Sie vom Programm gefragt, ob Sie dieses starten wollen:

DATAMAN S4 BIOS 1.1
128K BYTES RAM
START LIBARY ?

Mit **ESC**ape bewirken Sie einen Kaltstart mit den **BIOS** Werten.
Mit **ENTER** starten Sie das **LIBRARY**.
Wenn Sie ihr **LIBRARY** laden wollen, stecken Sie es in den **ZIF** Sockel und drücken Sie die Taste **LIB**.
Die Taste **LIB** bewirkt das Laden des **LIBRARY** Programms in den Transient Program Area von 8000 bis FFFF.
Ihr **LIBRARY** ROM muß dabei vom System als gültiges **LIBRARY** erkannt werden. Ein **ROM** ohne **LIBRARY** wird nicht geladen. Die **BIOS** Konfiguration des S4 verlangt einen 27256, deshalb muß ihr **LIBRARY ROM** ein 27256 sein.

Automatisches Abschalten

Wenn das Ladegerät nicht angeschlossen ist und 30 Minuten lang keine Eingabe erfolgt, schaltet sich der S4 selbstständig ab. Der S4 schaltet das Display ab und startet einen Abschalt-Modus. 30 Sekunden vor dem Abschalten macht er sich akustisch bemerkbar, danach schaltet der S4 aus. Wenn Sie während der letzten 30 Sekunden eine Taste drücken, wird das Abschalten abgebrochen. Beim Abschalten gehen keine Daten verloren, Sie müssen das Gerät jedoch **AUS**- und **EIN**- schalten um wieder arbeiten zu können.
Wenn das Ladegerät angeschlossen ist und die Ladung der Akkus unter 25% ist, wird diese automatisch geladen. Der S4 schaltet auch automatisch ab, wenn die Spannung der Akkus unter 8,4 Volt abfällt, da zu diesem Zeitpunkt zwar Daten und Programme gespeichert werden können, eine Eingabe aber nicht mehr möglich ist. Der S4 warnt Sie davor mit akustischen Signalen und einer Anzeige auf dem Display. Wenn die Spannung der Akkus

unter 8 Volt fällt, schaltet der S4 sofort, ohne Warnung, ab. Der S4 kann ohne Spannungsversorgung nichts sinnvolles mehr machen, außer den Speicher in Betrieb halten.

Die einzige Lösung dieses Problems ist es, die Akkus zu laden.

Speichererweiterung auf 512K

Der Speicher vom S4 kann durch auswechseln des **1M** Statik **RAM** Chip gegen einen **4M** Chip vergrößert werden. Der einzige verwendbare Chip ist der Hitachi HM628512LP.

Der Aufrüstvorgang ist sehr leicht:

1. Schalten Sie das Gerät aus.
2. Schrauben Sie die Schrauben an der Rückseite heraus und öffnen Sie das Gehäuse.
3. Entfernen Sie den **1M** Statik **RAM** Chip aus seinem Sockel und ersetzen Sie ihn durch den **4M** Chip.
4. Schließen Sie das Gehäuse und ziehen Sie die Schrauben wieder an.

Wenn Sie das Gerät jetzt einschalten, dann zeigt Ihnen der S4 jetzt 512k BYTES RAM an Stelle von 128k auf dem Display an.

Der ZIF Sockel

Der **ZIF** (**Z**ero - **I**nsertion - **F**orce) Sockel wird verwendet um neue **LIBRARY** Programme in den **TPA** (**T**ransient **P**rogram **A**rea) zu laden. Der **ZIF** Sockel wird auch dazu verwendet, um **PROMS** bei der Programmierung zu halten. Der S4 kann **PROMS**, so verwenden wie ein Computer Disketten: Sie sind Permanentspeicher Medien, welche Programme beinhalten, die in das System-**RAM** geladen werden können. (wie z.B. das **LIBRARY ROM**).

Wenn der **ZIF** Sockel vom S4 nicht angesprochen wird, liegt an diesem keine Spannung an. **PROMS** können jederzeit ausgewechselt werden, außer bei Operationen wie **LIB, LOAD, BURN, TEST, SUM** oder **COMP**, die von einem **PROM** lesen. Wenn der S4 auf Kommandos wartet, oder irgendwelche Funktionen ausführt, die nicht auf das **PROM** zugreifen, ist der Sockel nicht in Betrieb. Wenn der Sockel von S4 angesprochen wird, liegt nur Spannung an, solange von dem **PPROM** gelesen wird. **PROMS** mit 24 oder 26 Pins müssen unten, am Ende des Sockels, eingesteckt werden. Die oberen Pins des Sockels müssen frei bleiben.

Ausdrücke und Grundbegriffe

Die Nachricht **ESC** im Display meint, daß während eines Vorganges oder der Ausführung eines Kommandos, die **ESC** Taste gedrückt wurde. Das momentane Kommando wurde nicht vollständig ausgeführt oder noch gar nicht begonnen.

Die Nachricht **ZIF** meint den **Z**ero - **I**nsertion - **F**orce Sockel auf der Vorderseite.

Keypad bezieht sich auf die 45 Tasten von S4. Tastenwiederholung ist automatisch dann, wenn eine Taste längere Zeit niedergedrückt wird. Die Verzögerung bei der ersten Eingabe ist länger als bei Folgenden, um falsche Eingaben zu vermeiden. Wenn Daten wiederholt werden, wird der Blockcursor in einen Unterstrichcursor umgewandelt, um den Vorgang leichter verfolgbar zu machen.

Die **Screen** Nachricht meint das Remote Terminal Screen.

Die **Display** Nachricht meint entweder den Terminal Screen oder das S4 Flüssigkristall Display.

Die **Keyboard** Nachricht verweist auf das Remote Terminal.

Die **LCD** Anzeige ist ein **L**iquid **C**rystal **D**isplay. Die meisten Meldungen erklären sich von selbst, z.B.:

```
DATAMAN  S4
512K  BYTES  RAM
NO  LIBRARY  LOADED
>
```

Die **Command Line** ist die Display Zeile, die mit > beginnt.

Eine Operation wird ausgeführt, wenn Sie **ENTER** drücken oder abgebrochen beim drücken auf **ESC.**

Befehle, die nichts wesentliches bewirken, beginnen sobald eine Taste gedrückt wird, egal ob **ENTER** oder nicht.

Ein **Digit** ist immer Hexadezimal, nie Dezimal.

Eine **Adresse** definiert einen Platz im Speicher (ausgedrückt durch 5 Digits).

Ein **Parameter** ist eine Anordnung von **Digits**, zwei für ein **Byte**, fünf für eine **Adresse**.

Ein **Block** ist eine Ansammlung von **Bytes** im Speicher von einer **Startadresse** bis inklusive einer **Endadresse**. Wenn drei Parameter vorhanden sind, ist die letzte die **Bestimmungsadresse**.

Backspace und **Leertaste** werden zum Editieren von Parametern in der Command Line vom Terminal Keyboard verwendet, gleich wie Cursor Links/Rechts, auf dem Keypad.
ESCape wird verwendet um Kommandos abzubrechen, während diese bearbeitet werden.
Der Teil des Kommandos, der vor dem Drücken bereits ausgeführt wurde, kann dadurch nicht rückgängig gemacht werden.
ENTER (oder **RETURN**) startet Kommandos und Befehle.
ESCape oder **ENTER** können überall in der Kommandozeile gedrückt werden. Wenn die Parameter verändert wurden, kann mit **ENTER** bestätigt, oder mit **ESC**ape abgebrochen werden.
Wenn **drei Parameter** in der Kommandozeile stehen, dann sind sie in folgender Reihenfolge: **START**, **END**, **DESTINATION**. Jeder Parameter besteht aus fünf hexadezimalen Digits.

```
>
>LOAD  2764
00000-01FFF=04000
```

Die **START** und **END** Adresse gibt normalerweise die ganze PROM Definition an. Diese Adressen können editiert werden, um die Funktion von weniger als einem ganzen PROM zu verringern. Wenn Sie die **START** Adresse verändern und dabei die - Taste oder die ¯ Taste gedrückt halten, dann wird die **START** und die **END** Adresse automatisch auf die neuen Grenzen gesetzt. Wenn die **START** Adresse geändert wurde und über den vorherigen Grenzwert kommt, so wird automatisch die **END** Adresse auf den neuen Speicherbereich angepaßt. Wenn die **START** oder **END** Adresse vom Grenzwert weg verändert wird, dann wird das = Zeichen in der Kommandozeile in ein **#** Zeichen umgewandelt. Es ist dem S4 nicht möglich größere Speicherbereiche, als Arbeitsspeicher vorhanden ist, zu vergeben. Wenn das zu bearbeitende PROM größer als der vorhandene Speicher ist, erscheint vor dem **DEST** ein **#** Zeichen. In diesem Fall ist es nicht möglich, das ganze PROM in einem Durchgang zu bearbeiten. Die **DEST** Adresse müßte in so einem Fall mit den - **,**¯ Tasten in kleinere Blöcke unterteilt werden, die dann jeweils einzeln angesprochen werden.

```
>BURN
AMD/MMI  27C080
12.75V  TO  PIN  1
 00000-1FFFF#40000
```

Nur eine Grundeinstellung von **START**, **END** und **DEST** Adressen ist gespeichert und wird von allen Funktionen, die diese benötigen, verwendet. Wenn die **ESC**ape Taste während des Editierens gedrückt wird, wird zwar die Parameterzeile verlassen, aber die Werte für **START**, **END** und **DEST** werden nicht auf die ursprünglichen Werte zurückgesetzt.

Paged EPROMS (z.B. 27513)

Eine **Page** hat eine spezielle Bedeutung für **EPROMS**, wie z.B. den 27513, der in Pages von jeweils 16k Bytes unterteilt ist. Der S4 behandelt den Page Modus für EPROMS so, als wenn dieser über einen endlosen Speicher verfügen würden. Das Wechseln der Pages wird von S4 selbstständig übernommen, ohne den Benützer zu fragen.
Damit keine unterschiedliche Behandlung erfolgt, spricht der S4 27512 Bausteine mit einem 64K Bytes Block , 27513 Bausteine hingegen mit vier Pages a 16K Bytes an. Sie als Benutzer müssen sich über diese Unterschiede im klaren sein und wissen wie das System reagiert. Die genaue Benutzung dieser Teile ist in den Herstellerdatenblätter beschrieben.

Die FUNC Taste

Die **FUNC** Taste wird wie die Schift Taste bei einer Schreibmaschine benutzt. Man kann mit ihr eine Reihe von Extrafunktionen aufrufen. Wenn das System auf eine Funktion wartet, dann ändert sich die Eingabeaufforderung von > auf * .

Hörbare Töne

Der S4 macht verschiedene Arten von akustischen Signalen:

- Ein **einzelner** Ton wird gemacht, wenn eine Taste vom S4 angenommen wurde.
- Ein **doppelter** Ton, der auf und ab schwellt, ist ein Zeichen dafür, daß eine Taste nicht angenommen wurde.
- Während der Ausführung einiger Kommandos, die länger als 2 Sekunden dauern, wird vom S4 jede halbe Sekunde ein Ton erzeugt, um zu zeigen, daß das System arbeitet. Dies geschieht bei allen Funktionen außer

bei der **BURN** Funktion.
• Ein ankommendes Datenfile macht einen sehr auffälligen Sound.

Im **ADVANCED SETUP** kann den einzelnen Funktionen andere Erkennungstöne zugeordnet werden..

Computer Operationen

Es gibt 2 verschiedene Möglichkeiten den S4 zu verwenden. Die meist gebrauchte ist die, Kommandos über die Tasten einzugeben und von der LCD Anzeige zu lesen. Die andere Möglichkeit ist die, die RS232 Schnittstelle eines Computers zur Programmierung zu verwenden. Sie können dann über die Tastatur Daten eingeben und auf dem Bildschirm die Informationen lesen. Es ist ein Unterschied zwischen den Stand-Alone und den Remote-Operationen, aber jede Taste auf dem S4 hat eine gleichwertige Tastenklombination auf der Computer-Tastatur. Der S4 bearbeitet immer gleichwertige Signale. So zeigt er auf dem LCD Display Meldungen an, wenn mit den S4 Tasten eingegeben wurde, und auf dem Bildschirm, wenn die Information vom Keyboard gekommen ist. Wenn Sie während einer Operation die RS232 Schnittstelle abstecken, ist es notwendig den S4 erneut zu starten, um die Eingabe wieder über die S4 Tasten vornehmen zu können.

Wenn eine Funktion, entweder von der Tastatur oder mit den S4 Tasten, eingegeben wurde, wird das jeweils andere Eingabegerät, bis die Funktion beendet ist, ignoriert.

Verbinden mit einem Computer

Wahrscheinlich wird jeder S4 Benutzer einmal eine serielle Verbindung zwischen seinem Computer und dem S4 herstellen wollen. Jeder Computer besitzt normalerweise einen seriellen Port, über den Files mit anderen Geräten ausgetauscht werden können. Das Betriebssystem unterstützt normalerweise das Kopieren von Files auf serielle Geräte. MS DOS benutzt normalerweise COM1: oder AUX:, CP/M bevorzugt RDR: und PUN: z.B. das Kommando um mit einem IBM PC ein File zum S4 zu übermitteln ist:

COPY FILE.HEX COM1:

Die Baud - Rate, die Wortlänge, die Stop - Bits und die Parity Settings müssen aber vorher auf die richtigen Werte gesetzt werden. Das Kommando um dies zu machen ist:

MODE COM1:9600,N,8,1

Der S4 muß gleich eingestellt sein wie der Computer. Das File Format muß einem INTEL HEX File entsprechen, damit es vom S4 entschlüsselt werden kann. Wenn Files vom Computersystem überspielt werden, dann läuft dies ohne Geschwindigkeitsproblemen ab, da sich der S4 an die Geschwindigkeit des Rechners anpaßt. Beim Überspielen von Daten vom S4 auf einen Computer kann es allerdings zu Problemen kommen, da die meisten Computer in dieser Hinsicht nicht kompatibel sind. Am besten ist es für Sie, wenn Sie ein **Terminal** oder ein **COMMS Programm** verwenden.

Terminal Emulating Programme

Ein Terminal Programm sendet Informationen, die Sie übers Keyboard eingeben, an den Seriellen Port. Das Programm zeigt auch an, was über den Seriellen Port wieder zurück zum Computer kommt. Wenn der S4 mit Ihrem System verbunden ist, während Sie ein Terminalprogramm laufen haben, dann sieht es so aus, als ob alle Informationen, die Sie an den S4 senden sofort an den Bildschirm gegeben würden. Dies ist aber nicht so, sondern der S4 bekommt alle Informationen und sendet Sie wieder zurück an den Bildschirm. Ihr System funktioniert also über S4.

S4DRIVER

Dataman liefert gratis eine Terminal Software für Ihren PC zur Verwendung von S4. Der **S4DRIVER** beinhaltet Onlinehilfen und arbeiten mit 115200 Baud.

File Formate

Der S4 sendet und empfängt normale Computerfiles. Nur ein Programmierer mit viel Geduld würde es sich wünschen, den S4 mit endlos langen **HEX** Codes zu füttern. Microsoft hat in letzter Zeit ein Übersetzungssystem geschaffen, daß Microprozessorcodes in Maschinensprache übersetzt. Dadurch können Sie Zeit sparen, da Sie

das Programm nicht im **HEX** Code programmieren müssen. In letzter Zeit wird immer mehr mit einem Assembler programmiert, mit diesem können die Programmabläufe simuliert und übersetzt werden. Der Assembler erzeugt automatisch ein File im Maschinencode. Der S4 kann die gängigsten Fileformate verwenden, wie z.B. **INTELHEX**, **MOTOROLA S**, **TEKHEX**, **ASCII** oder **BINARY** und übersetzt diese in Objektcodes, die im **USER RAM** gespeichert sind.

Für den schnellen Anfang

Für alle, die sich die Bedienungsanleitung nicht durchlesen wollen.

Kopieren eines PROM

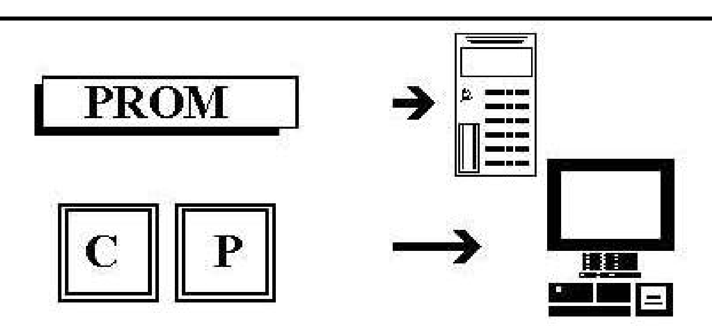
Wenn Sie Kopien von einem Master PROM, auf blanke PROMS der gleichen Serie erstellen wollen, dann geht dies in vier Abschnitten:

1. Konfigurieren Sie den S4 für diesen PROM Type.
2. Laden Sie das Master PROM in den S4.
3. Überprüfen Sie, ob das neue PROM wirklich leer ist.
4. Brennen Sie das neue PROM.

Konfigurieren eines PROM

1. Geben Sie das **MASTER PROM** in den **ZIF** Sockel, mit der Kerbe nach oben. Wenn das PROM weniger Beine hat als maximale Steckplätze vorhanden sind, positionieren Sie es ganz unten.
2. Drücken Sie **PROM**. Wählen Sie den Hersteller mit den - **,**¯ Tasten. Wählen Sie dann die richtige Stücknummer mit den ¬ , ® Tasten . Drücken Sie **ENTER**, wenn die richtige PROM-Nummer im Display erscheint. Wenn Sie ein Datenblatt haben, überprüfen Sie, ob die richtige Programmierspannung angezeigt wird.
3. Drücken Sie **LOAD**. Der S4 kopiert nun den ganzen **PROM** Inhalt in seinen Speicher. Wenn Sie das Programm nicht ändern wollen, drücken Sie **ENTER**, ansonsten können Sie jetzt die Parameter neu definieren.
4. Um zu überprüfen, ob die Daten richtig kopiert wurden, drücken Sie **COMP**. Der S4 vergleicht jetzt die Daten im **PROM** mit denen im **RAM**. Wenn dies korrekt erledigt wurde, erscheint die Meldung **SAME**.
5. Geben Sie ein neues, leeres **PROM** in den Sockel. Wenn es eine unterschiedliche Stücknummer hat, müssen Sie es neu konfigurieren.
6. Drücken Sie die Taste **TEST**. Der S4 wird Ihnen anzeigen, daß das **PROM** programmiert werden kann. Wenn die Nachricht „**WILL BURN**“ oder „**BLANK PROM**“ erscheint, dann ist alles in Ordnung. Anderenfalls müssen Sie das **PROM** erst löschen. **FLASH** Bauteile können mit **FUNC BURN** (zuerst **FUNC**, dann **BURN** drücken) gelöscht werden. **EPROMS** mit einem Fenster können mit einem UV Löschgerät gelöscht werden.
7. Drücken Sie zuerst **BURN** und dann **ENTER**. Die Parameterzeile kann verändert werden, wenn Sie zum Beispiel nicht das ganze **PROM** brennen wollen, was aber unüblich ist. Die Zeit, die ein **PROM** braucht, um gebrannt zu werden, schwankt von einigen Sekunden bis zu ein paar Minuten.
8. Es ist ratsam sich die **CHECKSUM** von einem **MASTER PROM** und den Kopien aufzuschreiben. Die **CHECKSUM** hilft Ihnen, Ihre unmarkierten **PROMS** zu identifizieren.

PROM Taste



Wenn Sie beim Drücken dieser Tasten die folgende Meldung erhalten:

```
>SELECT  DEVICE
NO  LIBARY  LOADED
>
```

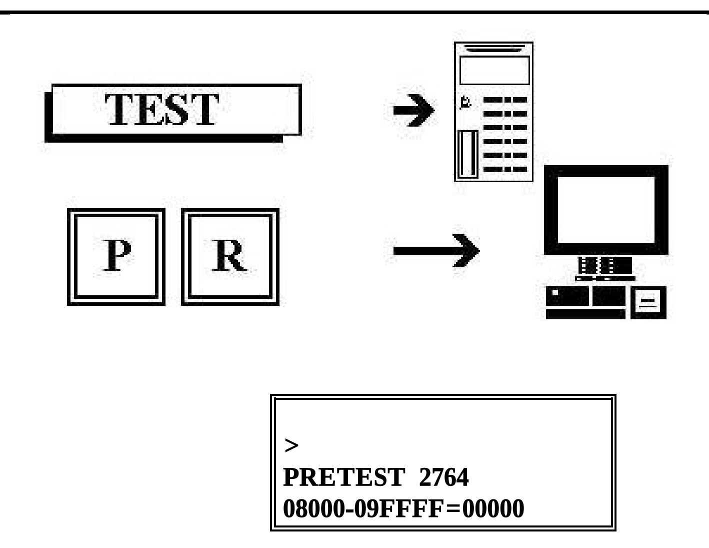
Dann kann S4 nur 27256 **PROMS** programmieren. Sie sollten unbedingt ein **LIBRARY** laden.
(Siehe **LIB Taste**)

```
ATMEL  27C010
13:00V  BURN PIN 1
100US  NO  OVERPROG
ALG  01,  REF  0400
```

Um den S4 richtig zu konfigurieren, damit dieser **PROMS** richtig ließt und programmiert, beginnen Sie wie folgt:

1. Wenn Sie die Ref Nummer des Algorithmus kennen, geben Sie diese ein. Wenn nicht, die Nummer wird angezeigt, wenn Sie durch den Algorithmus scrollen (genaue Beschreibung folgt). Die Ref Nummer ist in verschiedenen Versionen des **LIBRARYS** nicht unbedingt gleich.
2. Benützen Sie die - und ¯ Tasten, um durch eine Liste zu scrollen. Um das aktuelle PROM zu wählen, benützen Sie die ¬ und ® Tasten (Auf einem Terminal benützen Sie **X, Y, BACKSPACE, Leertaste**). Sie sollten die Teilenummer und auch die Programmierspannung des PROM wissen. Wenn Sie unsicher sind, benützen Sie ein Datenblatt des Herstellers.
   Der S4 legt die Programmierspannung an das PROM an, die im Display angezeigt wurde. Falls diese falsch ist, kann es zur Zerstörung des PROM kommen. Wenn Sie sich mit der Spannung nicht sicher sind, steigern Sie langsam die Spannung vom niedrigsten Wert an aufwärts.

TEST Taste

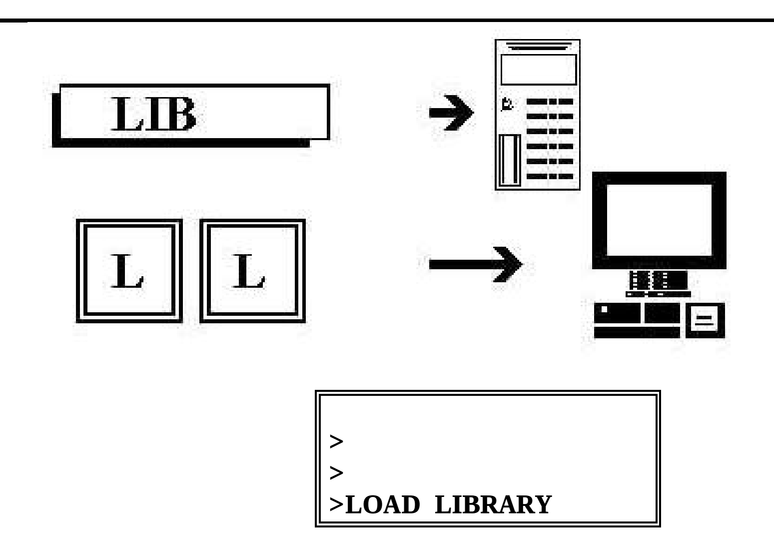



```
>
PRETEST  2764
08000-09FFFF=00000
```

PRETEST (START) - (END) = (DEST)
PRETEST vergleicht ein **PROM** im **ZIF** Sockel mit dem Inhalt des **USER** - **RAM's** zwischen der **START** und der **END** Adresse. Der Sinn liegt darin, zu überprüfen, ob das **EPROM** noch gelöscht werden muß. Die erste Adresse, die nicht programmiert werden kann, wird angezeigt. Wenn ein **PROM** Adressen beinhaltet, die bereits programmiert sind, das neue Programm aber Platz hat, dann wird die Nachricht **WILL BURN** im Display angezeigt. Wenn das **PROM** leer ist, d.h. wenn alle FF Bytes des **PROM** leer sind, dann erscheint die Nachricht **BLANK ROM** im Display. Der S4 muß als erstes für die richtige PROM Type konfiguriert werden. Ein Wahrnhinweis gibt die **PROM** Type an. Um das **PRESET** zu starten, drücken Sie die **ENTER** Taste.

Folgendes Beispiel: Machen Sie einen **PRESET** mit einem 2764, von den Adressen **00000** bis **01FFF** um zu sehen, ob das Programm des **USER RAM**, mit den Adressen **08000** bis **09FFF**, richtig programmiert werden kann.

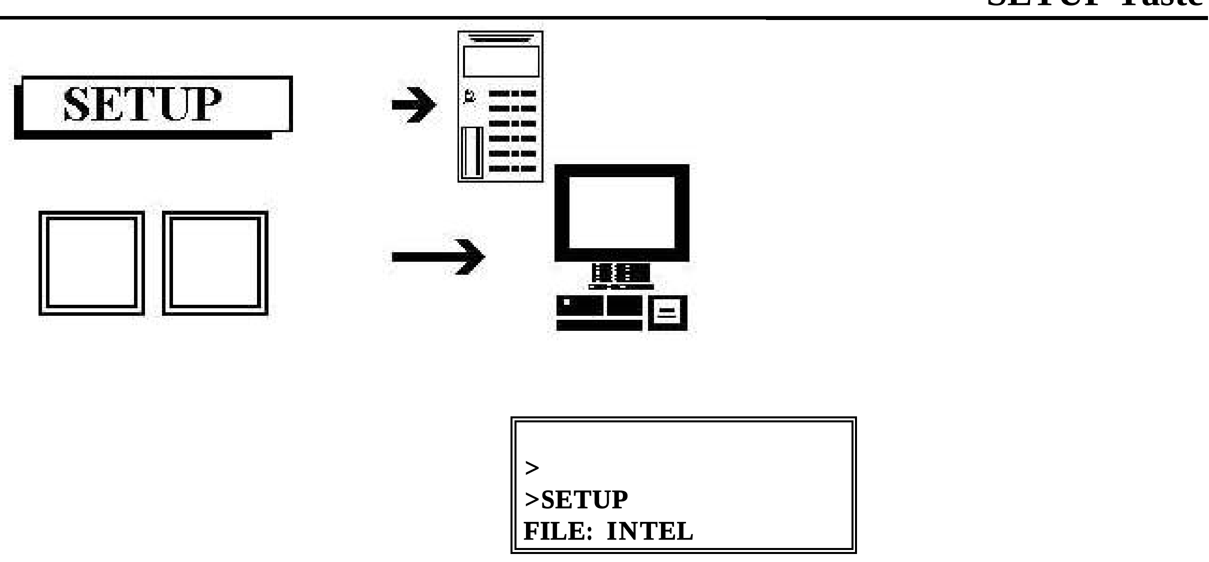
LIB Taste




```
>
>
>LOAD  LIBRARY
```

LIB wird gebraucht, um eine neue Software von einem **LIBRARY PROM** im **ZIF** Sockel, in das Transient Program Area zu kopieren. Das Programm läuft automatisch, nachdem es geladen wurde. Wenn Sie die Nachricht, **PLACE A LIBRARY ROM** sehen, dann können Sie **ENTER** drücken um zu bestätigen, oder **ESC**ape um abzubrechen. Der **LIBRARY** Befehl kopiert nicht alle Daten des **LIBRARY PROM** in den **TPA**, wenn es das machen würde, dann würde der Stack überschrieben werden und das Programm dadurch abstürzen. Statt dessen beinhaltet das **PROM** Pointer, die dem Programm zeigen, welche Blöcke verschoben werden müssen. Jeder Pointer ist beschrieben durch einen 42 Byte: 42 wird verwendet um anzuzeigen, daß ein Block vorhanden ist, der kopiert wird (Warum 42 ? Weil theoretisch irgendein Byte verwendet werden könnte. 42 aber sonst keine wirklich wichtige Bedeutung hat, außer für Software Ingenieure, die Douglas Adams Fans sind...). Die folgenden vier Bytes geben die **START** und die **END** Adresse des Blocks an. Wenn ein zweiter Block folgt, dann wird dieser wieder durch ein 42 Byte, eine **START** und eine **END** Adresse gekennzeichnet. Wenn ein **LIBRARY** Programm geladen wurde, startet der S4 automatisch neu, und beginnt das neue Programm. Am Ende der Block Tabelle ist ein Byte, daß ungleich 42 ist. Wenn dieses Byte 00, 54 ist, dann benutzt der S4 diese Bytes als Startadresse und beginnt mit einem Neustart. Wenn das letzte Byte nicht 00, 54 ist, dann kehrt der S4 zum Kommando **PROMPT** zurück.

SETUP Taste




```
>
>SETUP
FILE:  INTEL
```

Diese Taste hat keine gleichwertige Tastenkombination auf dem Terminal. File-Typ, Baud Rate und Handshake haben verschiedene Terminal Kommandos. Die **SETUP** Routine setzt den **FILE TYPE RS232, BAUD RATE** und **HANDSHAKING**, in dieser Reihenfolge. Benützen Sie die ¬ und die ® Tasten um die Optionen auszuwählen. Es ist möglich, jederzeit mit **ESC**ape abzubrechen, die umgeänderten Werte, werden aber nicht auf die Originalwerte zurückgesetzt. Vergessen Sie nicht, was Sie auf dem Display lesen, ist bereits geschehen. Es ist daher ratsam, die **BAUD RATE** nicht umzuändern, wenn Sie das **FILE FORMAT** ändern wollen.

FILE Typen

Der S4 kann Files, die seriell übermittelt werden, in allen Standartformaten empfangen. Die unterstützten Formate sind: **INTEL, MOTOROLA, TEKHEX, ASCII** und **BINARY.** Eine genauere Beschreibung jedes Formates folgt im Anschluß.
Mit den ¬ **,** ® Tasten können Sie durch die verschiedenen Formate scrollen, das unterlegte Format können Sie mit **ENTER** auswählen, oder mit **ESC**ape den ganzen Vorgang abbrechen.

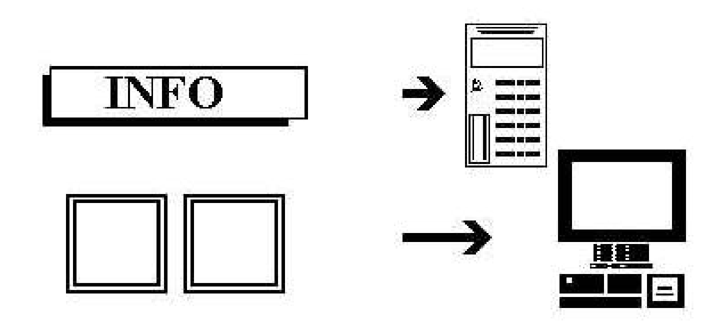
BAUD RATE & HANDSHAKE

BAUD RATEN von 300, 600, 1200, 2400, 4800, 9600, 14400, 28800 oder 115200 sind möglich.

FILE: INTEL
BAUD: 9600
AUTO - HIT SPACEBAR
HANDSHAKE: NONE

Mit den ¬ **,** ® Tasten können Sie durch die verschiedenen **BAUD RATEN** scrollen. Ihre Wahl können Sie mit **ENTER** bestätigen, oder das ganze mit **ESC**ape abbrechen. Eine automatische Erkennung der **BAUD RATE** ist auch möglich, wenn Ihr Computer mit einem **COMMS** Programm arbeitet. Schließen Sie dazu Ihren S4 an den Computer an und drücken Sie auf Ihrem Computer die Leertaste. Wenn ein **HANDSHAKE** beim Datenübermitteln erforderlich ist, kann dieser auf **RTS** oder **DTR** gesetzt werden.
Wenn der S4 ausgeschalten wird, wird das **HANDSHAKE** auf **NONE** gesetzt, beim Einschalten ist also kein **HANDSHAKE** mehr vorhanden. Anderenfalls würde der S4 abstürzen, da dieser auf eine Datenübertragung wartet.

INFO Taste



E ½½½½½½½½½½½½½½½ < F
BATT 8.8V, 24°C
CHARGER DISCONNECTED
OR SUPPLY OFF

Für diese Funktion gibt es ebenfalls keine gleichwertige Tastenkombination auf dem PC. **INFO** zeigt Ihnen den Ladungszustand und die Temperatur der Akkus an, und zeigt Ihnen auch, ob das Ladegerät angesteckt und eingeschalten wurde. Das Ladegerät funktioniert normalerweise selbstständig und bedarf keiner Überwachung. Der S4 wird mit wiederaufladbaren Akkus, bestehend aus 7 NiCd Zellen, welche 8,4 Volt bei 600 mAh liefern,

geliefert. Das Laden erfolgt automatisch, Sie müssen das Ladegerät also nicht **EIN** und **AUS** schalten. Das Ladegerät schaltet sich automatisch ein, wenn der Ladungszustandsmelder unter 25% fällt. Sie müssen das Ladegerät also nicht selbstständig ein und aus schalten, können dies aber in **INFO** Menü mit den ¬ **,** ® Tasten machen. Wenn das Ladegerät angesteckt ist, können Sie jetzt sehen, wie die Akkus geladen werden. Wenn die Batterie komplett geladen wird, steigt die Temperatur. Wenn der Temperaturanstieg ungefähr 5°C entspricht, schaltet das Ladegerät ab. Einmaliges Laden ist genug. Sie können die Batterie zwar mehrmals hintereinander laden, die maximale Ladung ist aber schon beim ersten Mal erreicht.
Um den **INFO** Modus zu beenden, drücken Sie die **ESC**ape Taste.

Automatisches Laden der Akkus

Wenn der S4 eingeschalten ist und der Ladungszustandsmelder unter 25% fällt lädt es die Akkus. Das Ladegerät kann die Batterie in weniger als einer Stunde aufladen. Ein Temperatursensor, in Kontakt mit den Akkusüberwacht den Temperaturanstieg beim Laden. Wenn dieser größer als 5°C ist, dann wird der Laadestrom abgeschalten. Das Laden der Akkus ist komplett sicher. Sie können den S4 während des Ladens wie gewohnt benutzen. Eine interne Schaltung schützt die Akkus vor Überladung. Mit einem Kondensator wird der überschüssige Strom gespeichert, was einen Temperaturanstieg zur Folge hat. Wenn dieser Temperaturanstieg 5°C übersteigt, schaltet der S4 den Ladevorgang ab. Wenn während des Ladens im Display die Nachricht **HEAT** erscheint, dann ist die Temperatur entweder unter 5°C oder über 45°C, sie kann auf jeden Fall nicht mehr überwacht werden. Die Temperatur wird mit Hilfe des Stroms, der durch den Leiter fließt, gemessen. Der Sensor ist in direktem Kontakt mit der Batterie. Es ist zwar kein wirkliches Thermometer, aber trotzdem gut genug um einen 5°C Temperaturunterschied festzustellen. Der S4 unterstützt das Laden der Batterie nicht, wenn die Temperatur außerhalb der vom Hersteller angegebenen Grenzen (5°C und 45°C) liegt.

Der Ladungszustandsmelder FUEL GAUGE

Am oberen Rand des Displays ist eine **BAR - Graph** Anzeige, die den Ladungszustand anzeigt. Es ist nicht genau möglich, den Ladungszustand eines Akkus durch Messen der Spannung festzustellen. Deshalb speichert der S4 die Ladungszustände beim letzten Laden. Der **FUEL GAUGE** ist eine Schätzung vom S4, wieviel der Akku geladen ist. Der Ladungszustandsmelder funktioniert in folgenden Situationen nicht ordnungsgemäß:

1. Wenn die Akkus ausgewechselt wurden.
2. Wenn der S4 über einen längeren Zeitraum hin ausgeschalten war (da der S4 die Selbstentladung der Akkus nicht berücksichtigt).

In allen Fällen kann der Ladungszustandsmelder wieder korrekt funktionieren, wenn die Akkus geladen werden.

NiCd Akkus

Die Akkus sind zu den folgenden typischen Werten fähig, wenn sie vollständig geladen sind.

Standby 10µA ungefähr 12 Wochen Erhaltung der im RAM gespeicherten Programme, der Daten und der Konfiguration. Nach dieser Zeit übernimmt die Lithium Zelle die Versorgung.

Viewing 30mA 20 Stunden lang Eingaben (25% weniger, wenn RS232 benutzt wird)

Burning 180mA 3 Stunden programmieren der aktuellen Nummern von PROM Vertretern, ungefähr 100 ältere und 1000 neue Typen.

Emulating 100mA 6 Stunden Emulation (abhängig von den Zugriffen, die das System auf den S4 macht und dem Load Platz in der Datenzeile).

Die Richtige Arbeit ist eine Kombination aus all diesen Tätigkeiten. Frühwarnsystem der Akkuladung und automatisches Ausschalten, wenn die Akkuladung unter 8,4 Volt fällt. Abschalten ist eine Vorsichtsmaßnahme um alle bereits programmierten Daten zu erhalten, die nach dem Laden immer noch vorhanden sind. Wenn das Ladegerät eingeschalten ist, fließen 650 mA AC und laden die Akkus in ungefähr einer Stunde. Die Mindestarbeitsspannung ist bereits in kürzerer Zeit erreicht. Fast vollständiges Entladen und anschließendes schnelles Laden garantieren ein längeres Leben der Akkus als immer wieder kurzes Laden.

Lithium Backup Batterie

Nickel Cadmium Akkus verlieren ihre Ladung bei einer Selbstentladung von 200% pro Jahr (Angabe des Herstellers). Das heißt, daß eine vollständig geladene NiCd Akku über 6 Monate lang nicht geladen werden muß. Wenn der S4 sich selbst abschaltet, weil die Spannung der Akkus zu niedrig ist, ist da immer noch 1% der Spannung vorhanden, um die Daten im RAM für ein paar Tage zu erhalten. Für diesen Fall ist die Lithium Batterie eingebaut worden, um die RAM und die Mikroprozessor Daten zu erhalten, wenn der NiCd Akku leer ist. Die Lithium Batterie hat eine sehr niedrige Selbstentladung, die Batterie hält deshalb ca. 10 Jahre in Ihrem S4, der immer geladen wurde. In einem S4, der nicht ständig geladen wurde, hält Sie ungefähr 2 Jahre.

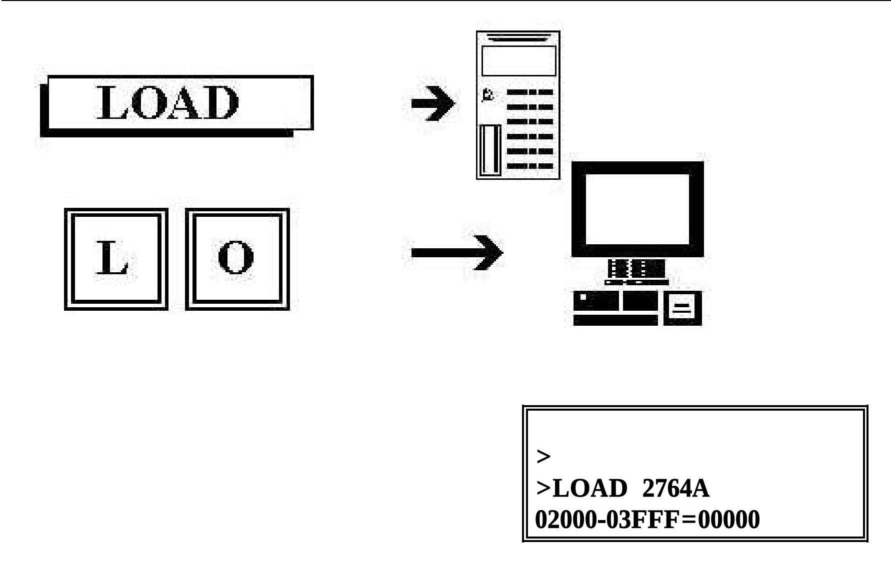
Laden mit einer DC Quelle

Sie können den S4 auch mit Gleichstrom laden, müssen dabei aber berücksichtigen, daß der S4 Gleichspannung nicht abschalten kann, da der Temperaturfühler nicht ordnungsgemäß funktioniert. Sie müssen also eine kleinere Spannungsquelle benützen, mit ungefähr 60 mA, damit sich die Batterie nicht überhitzt. Die Polung ist nicht wichtig, da im S4 ein Gleichrichter enthalten ist.
Vermeiden Sie es, den Emulationsmodus gleichzeitig mit dem Laden von einer Gleichspannungsquelle durchzuführen. Da es einen Konflikt mit der Systemmasse geben kann.

Fehlerhafte Batterie

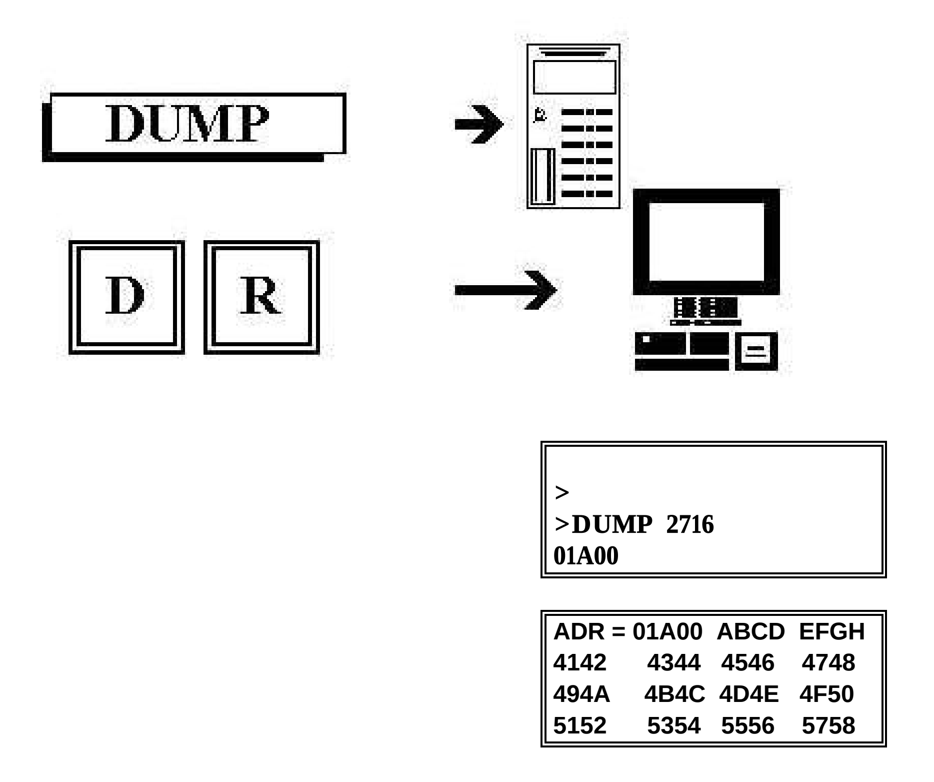
NiCd Akkuhersteller empfehlen das „**DEEP CYCLING**“, welches bedeutet, die Akkus fast vollständig zu entladen und anschließend wieder zu laden, ist. Elektronisches Gesetz ist, daß Ni-Ionen die Ladung lernen, die ihrer Anforderung entspricht. Einige Wissenschaftler behaupten, daß dies nicht so sei. Dataman hat niemals solche Fälle feststellen können. Das einzig feststellbare war, daß das Alter eines Akkus nicht mit der Ausfallswahrscheinlichkeit zusammen hängt. Neue Akkus fallen genauso oft aus wie ältere. Der einzige Grund dafür ist die sachgemäße Ladung der Akkus. Die Symptome eines defekten Akkus sind, z.B. daß diese nicht mehr die maximale Outputspannung erreicht. NiCd Akkus fallen aus, indem die Spannung, auf die maximal geladen werden kann, immer niedriger wird. Wenn eine der 7 NiCd Zellen nicht mehr komplett funktioniert, dann wird der ganze Akku beim Laden überhitzt, und kann, da nurmehr 6 Zellen vorhanden sind, nicht mehr die volle Leistung bringen. Die Kombination von hoher Temperatur und niedriger Betriebsspannung kann im **INFO** Fenster geprüft werden. Wenn dies der Fall ist, sollte die Akkus unbedingt durch Neue ersetzt werden.

LOAD Taste



LOAD (START) - (END) = (DEST)
Wird verwendet um den Inhalt eines **PROMS** in den **USER RAM** zu laden. Der Bereich der kopiert werden soll, ist mit der **START** und **END** Adresse definiert. Der S4 muß für die richtige **PROM** Type erst konfiguriert werden. Das Laden des Speichers benötigt keine Programmierspannungsimpulse zum **PROM. PROMS** mit der gleichen Größe und von der gleichen Type haben unterschiedliche Konfigurationen für Programme. Diese können aber mit der gleichen Prozedur gelesen werden. Wenn der einzige Unterschied darin liegt, daß das PROM von einem anderen Hersteller ist. D.h., wenn Sie ein **27256 PROM** auf ein gleiches, nur von einem anderen Hersteller, leeres **PROM** kopieren wollen, dann können Sie die Konfiguration, für das gerade bearbeitete **PROM**, beibehalten. Sie müssen die Konfigurationen nicht ändern, nur um das Original zu lesen. Ein Warnhinweis gibt den **PROM** Typ an. Wenn **ESC**ape gedrückt wird, wird das Programm unterbrochen, bei Druck von **ENTER** wird es fortgesetzt.
Folgende Aufgabe: kopieren Sie einen 2764, der im **ZIF** Sockel ist, mit den Adressen **00000** bis **01FFF** ins **USER RAM** auf die Adressen **02000** bis **03FFF**.

DUMP Taste



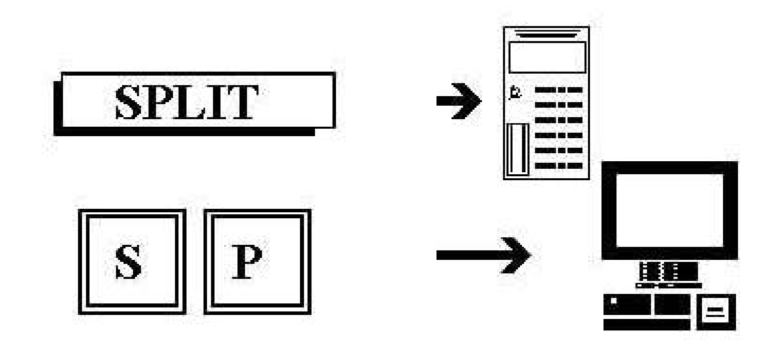
Die Keypad Version von dieser Funktion ist nur um zu editieren und wirkt direkt auf den Bauteil im ZIF Sockel wirkt. Für eine einfache Nutzung gibt es eine Editierroutine. Sie können keine der Daten ändern.

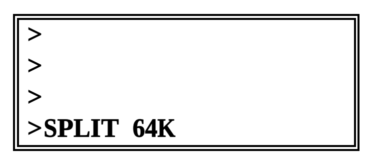
Terminal DUMP ROM

DUMP (DEVICE)
(START), (BYTES)
Das ist die Terminal Version, die gleich der Keypad Version ist. Es benutzt einen kompletten Bildschirm um die Datensätze anzuzeigen. Beginnend mit der benötigten Adresse für die Nummer von Bytes. Wenn die Nummer von Bytes auf 0 gesetzt wurde, setzt **DUMP** die Funktion fort, bis **ESC**ape gedrückt wird.

*Achtung: Computer können lange brauchen, bis ein Bildschirm mit Daten vollgeschrieben ist. Unbegrenzte DUMPS können den Input Buffer des Computers zum überlaufen bringen. Sie merken dies an einem fragmentierten Birdschirmaufbau. In so einem Fall könnte die Ursache bei einer zu hohen **BAUD RATE** liegen.*

SPLIT TASTE





Gibt alle ungeraden Bytes in die obere Hälfte des Speichers, und die geraden Bytes in die untere Hälfte des Speichers. **SPLIT** teilt alle 64K, (außer den Positionen 0000 und FFFF) auf. Verwenden Sie die ¬ und die ® Taste um den Speicher von 128K, 256K oder 512K Bytes aufzuteilen. Wenn ein Microsystem einen Bus mit 16 oder 32 Bit Breite hat, dann kann dieses mehr als nur ein **PROM** Parallel adressieren. Aber ein Assembler produziert den Code in Serie, mit abwechselnden High und Low Bytes. Es ist möglich, daß der S4 den seriellen Code empfängt und jedes einzelne Byte in ein anderes **PROM** schicken muß. Das Aufteilen des Codes macht es notwendig, daß einzelne Bytes zu einem Block im Speicher gebunden werden, und dieser sofort auf das **PROM** geschrieben wird.

Wie die Speicheraufteilung arbeitet

Beispiel: Umwandeln eines 64k x 8 Files in ein 16k x 32

Load 64k

FFFF

0000

Split memory

FFFF

ODD adress code

8000

EVEN adress code

0000

Split memory again

FFFF

Sector 3 = 0003, 0007 & c. — **C000**

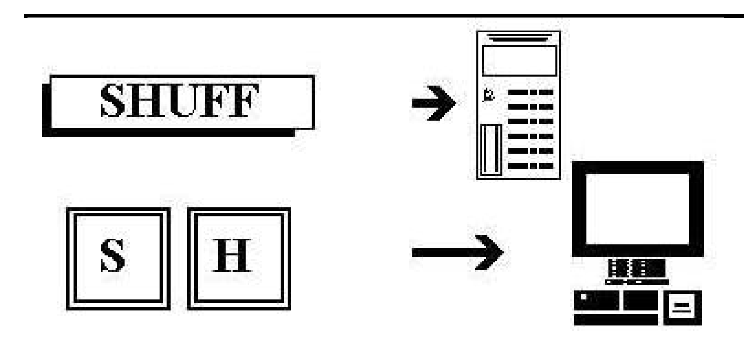
Sector 2 = 0002, 0006 & c. — **8000**

Sector 1 = 0001, 0005 & c. — **4000**

Sector 0 = 0000, 0004 & c. — **0000**

Program SET of four 27128 with sectors 0,1,2,3

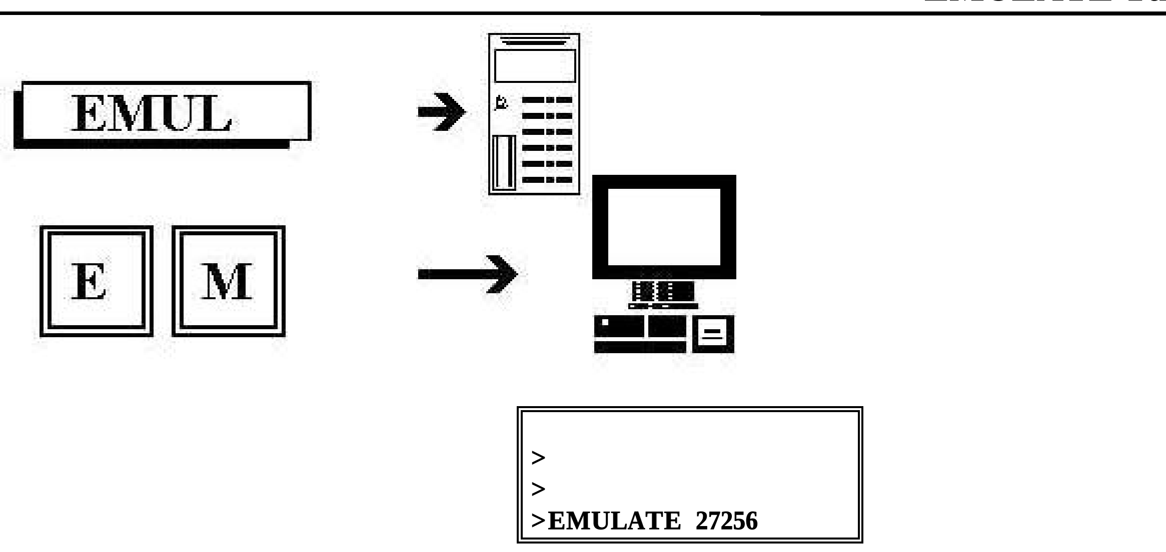
SHUFFLE Taste



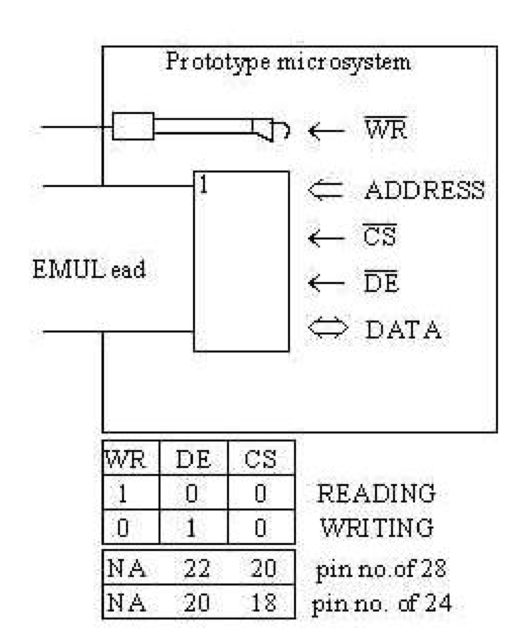


SHUFFLE ist das Gegenteil von **SPLIT**. Gibt alle ungeraden Bytes in die obere Hälfte des Speichers, und die geraden Bytes in die untere Hälfte des Speichers. Jedes Byte im Speicher wird bewegt, außer dem ersten und dem letzten - 0000 und FFFF. Verwenden Sie die ¬ und die ® Taste um den Speicher von 128K, 256K oder 512K Bytes zu **SHUFFLE'N**. Wenn **PROMS** ins **USER RAM** geladen werden und der Code zu einem Terminal, über eine **RS232** Schnittstelle, geschickt wird, kann dies in einem 16 oder 32 Bit System erfolgen. Die Bytes müssen verbunden sein, damit der Computer diese erfolgreich lesen kann. Dieses kann erleichtert werden, indem man die verschiedenen **PROMS** in unterschiedliche Blöcke im Speicher lädt, und anschließend den Befehl **SHUFFLE** verwendet.

EMULATE Taste



Emuliert das konfigurierte **PROM**. Wenn der **WRITE** Draht mit dem Mikroprozessor verbunden ist, kann aus dem **USER-RAM** über das **EMU**-Flachbandkabel durch das Kabel geschrieben werden. ESCape bricht die Emulation ab und kehrt zur **PROMPT** zurück (Alle anderen Tasten bewirken das gleiche).



Speicher Emulation

Der S4 emuliert entweder **RAM** oder **ROM**, und kann zur Modifikation oder zur Neuentwicklung benutzt werden. Das Emulationskabel sollte im Trägersystem, entweder in den **RAM** oder den **ROM** Sockel gesteckt werden, bevor das Emulationsprogramm gestartet wird. Die Pin-Out Konfiguration von **EPROMS** erfolgt automatisch, **Chip Enable** und **Output Enable Signale** sind richtig eingebaut für **EPROMS** der **JEDEC 27** Serie. Diese Technik ist eine Weiterentwicklung der **ROM Emulation**, genannt **Speicher Emulation**.

EMULation von der Adresse 0

Das Emulationskabel bewirkt, daß sich das S4 **USER RAM** wie ein **PROM** verhält. Das Trägersystem sieht die Nulladresse des emulierten **PROM** an der Nulladresse im S4 **USER RAM**. Wenn die Emulation in einem höheren Adressbereich beginnt, dann muß es auf die Nulladresse vom **USER RAM** geschoben (**geswapped**) werden, wenn man die Emulation startet. Die Pins des Emulationskabels sind mit dem **USER RAM** über den Buffer verbunden. Wenn die 00000 Adresse, die Chip-Select und die Output Enable Signale auftreten, dann wird die 00000 Adresse des **USER RAM** auf den Datenbus geschrieben. **PAGED** Adressing, wie es in den 27513 **EPROMS** verwendet wird, bei denen andere **PAGES** 0 sind, wird nicht unterstützt.

*Es kommt öfter vor, daß der S4 nicht in den Emulationsmodus schalten kann, weil der Benutzer z.B. einen 2764 an der Adresse **0E000 - 0FFFF** geladen hat, da dies die Adresse ist, die er in seinem Prototyp gespeichert hat. Ein **PROM** weiß nicht, wo in seinem Speicher die gesuchten Adressen liegen, genauso ist es beim S4, wenn dieser emuliert, da das Decodieren im Trägersystem erfolgt. In diesem speziellen Fall würde es notwendig sein, den Code von **0E000** auf **00000** zu schieben, bevor mit dem Emulieren begonnen wird.*

Vorteile der Speicher EMULation

1. Es ist universell einsetzbar. D.h. Sie können es mit jedem Mikroprozessor benützen.
2. Die Ausrüstung kostet weniger, als ein Mikroprozessor Entwicklungssystem. Die einzige zusätzliche Ausrüstung, die Sie benötigen, ist ein Computer und einen **CROSS ASSEMBLER**.
3. Das Träger System verhält sich wie das richtige **EPROM**.
4. Die Software läuft mit voller Geschwindigkeit.
5. Speicherinhalte können angesehen und editiert werden, anschl. kann das Programm wieder schnell gestartet werden.
6. Es ist sehr billig und effektiv mit **SINGLE CHIP MICROCONTROLLERN**, die eine Piggy-Back Version besitzen. Sie müssen nur an der hinteren Seite des **MICROPROZESSORS** das Kabel einstecken.
7. Es ist gut geeignet um Modifikationen bei bereits existierenden Systemen durchzuführen. Um z.B. Hinweise und wichtige Nachrichten zu ändern, wenn Fehler angezeigt werden. Das Ausprobieren von neuen Versionen und oftmaliges Wiederversuchen funktioniert mit dem S4 sehr gut.
8. Wenn das Programm läuft, kann sofort das passende **EPROM** dazu gebrannt werden.

Wie die Speicher EMULation arbeitet

Es wird ein Programm auf einem Computer in einem der Standart Fileformate in den Emulator geladen. Der Speicher Emulator macht dabei die meisten Dinge, die auch ein **M**ikroprozessor **E**ntwicklungs **S**ystem (**MES**) machen kann, obwohl die meisten der **MES** nicht so viele wirksame Editoren und Code Manipulatoren gespeichert haben, wie der S4. Der S4 kann nicht alle Dinge so gut erledigen wie ein echtes **MES**, aber es ist normalerweise nicht zu schwer, eine Routine in dem Trägermikroprozessor Code zu schreiben, wobei vielleicht ein Subroutine Aufruf oder ein Software Interrupt durch eine Anwendung ersetzt wird. Dabei wird das Mikroprozessor Register in einen bestimmten Bereich des **USER RAMS** geladen, wo es kontrolliert werden kann. Der Mikroprozessor im Träger System kann sowohl Schreiben als auch Lesen. Es ist richtig, daß **PROMS** keinen Schreibeingang haben, aber das kann verhindert werden, indem eine separate Leitung an den **WRITE** Draht angehängt wird.

Microsystem Speicher Auswahl

In einem Microsystem werden **ROM** und **RAM** mittels Adreßbus decodiert und anschl. mit dem **CHIP SELECT** Signal ausgewählt. Jeder Chip hat seinen eigenen Slot im Adressierbereich. **ROM** und **RAM** geben ihre Daten frei, wenn das Bussystem ihre Adresse auswählt und die **READ** Leitung vom Mikroprozessor auf „ true “ geschalten wird. Die **READ** Leitung sollte mit den **OUTPUT ENABLE** Leitungen, aller Chips in diesem System, verbunden sein. Der Mikroprozessor kennt während dem Lesevorgang keinen Unterschied zwischen **ROM** und **RAM**. Während einem **WRITE** Zyklus wird die **READ** Leitung auf „ false “ geschalten und kein

Speicherchip kann seine Daten in den Bus abgeben. Das verhindert Konflikte mit dem Mikroprozessor, wenn dieser Daten auf den Bus schreibt. Der **WRITE** Ausgang des Mikroprozessors ist mit dem **WRITE** Eingang des **RAM** verbunden. Wenn das **WRITE** Signal „ true “ wird, dann übergibt der Mikroprozessor seine Daten an das **RAM**, wobei die einzelnen Sektionen mit dem Adreßbus zugewiesen werden. Der Mikroprozessor würde auch auf das **ROM** schreiben, dies wird aber dadurch verhindert, daß dem **ROM** der **WRITE** Eingang fehlt.

Korrektes Prototypen Design

Es gibt viele Mikrosysteme, die vollkommen falsch entworfen sind. Ein häufiger Fehler ist, daß die Speicherchips meistens über ihre **OE** und nicht über ihre **CS** Eingänge angesprochen werden. Ein anderer Fehler besteht darin, daß die **OE** und **CS** Eingänge miteinander verbunden werden, oder gar auf Masse gelegt werden. **CS** sollten nämlich mit dem **ADRESS BUS** und **OE** mit dem **READ STROBE** verbunden sein. Manche Systeme werden auch mit mehr Strom betrieben, als notwendig wäre. Es gibt eine Periode von Konflikten in jedem Zyklus, in welchem beide, der Mikroprozessor und die Speicher Chips, berechtigt sind auf den Bus zuzugreifen. Dies verursacht kurze Strom- und Spannungs Spitzen. Manche Speicher Teile benötigen längere Zugriffszeiten, um sich von so einem Konflikt zu erholen. Die Zugriffszeit ist auch dann länger, wenn das System keine **CS** Hauptleitung oder **OE** Leitung hat. Zugriffszeiten unter **OE** sind immer kürzer. **ROM** Emulationen arbeiten in solchen Systemen, aber **RAM** Emulationen können in solchen Systemen nicht funktionieren, weil es kein **CS** ohne **OE** verarbeiten kann. Noch weniger würde dies mit einem echten **RAM** funktionieren. **CHIP SELECT** muß „ true“ und **OUTPUT ENABLE** muß „false“ sein während des Schreibvorganges.

RAM Emulation

Byte-Wide RAMS können auch emuliert werden, wenn die Emuleitung in den Sockel für das **STATIC RAM** gesteckt wird, dabei muß das **PROM** aber so konfiguriert sein wie ein **PROM** mit gleichen Adressen und Data Pin Outs. Die **WRITE** Leitung darf nicht über das **EMU** Kabel verbunden werden, sondern es muß direkt zum Flying **WRITE** Draht gehen. Eine Möglichkeit besteht darin, die **WRITE** Leitung am Stecker abzuschneiden, und das Kabel nur für diesen Zweck zu verwenden. Eine andere Möglichkeit ist, sich gleich ein anderes
EMU Kabel zu kaufen, bei dem bereits der **WRITE** Draht weggelassen wurde, und der **WRITE PIN** fehlt.

Leistungsverbrauch beim Emulieren

Die Zeitausschaltung ist beim emulieren außer Betrieb, die **LOW Batterie** Abschaltung funktioniert aber weiterhin. Die Emulation stellt für die Akkus eine eigene Herausforderung dar, denn der normalerweise fließende Emulationsstrom liegt bei 100mA. Bei dieser Belastung halten die Akkus ungefähr 6 Stunden, bevor sie wieder neu geladen werden müssen. Wenn Sie das Netzgerät eingesteckt haben, dann besteht natürlich keine Zeitbegrenzung. Mit **ESC**ape beenden Sie den Emulationsmodus.

Drahtenden der EMUL Leitung

Alle **DATA**, **ADDRESS** und **CONTROL** Drähte sind über logische **74HCT** Gatter, mit einem Serienwiderstand von 150Ω, mit dem S4 verbunden. Diese minimalen Abweichungen bewirken einen **OVERDRIVE**. Um zu vermeiden, daß beim Ausstecken hohe Stromspitzen auftreten, haben alle Drähte über einen 100kΩ Widerstand eine Verbindung auf Masse oder 5 Volt. Um Stromspitzen während der **EMUL**ation zu vermeiden sollten die **INPUTS** zwischen 0 und 5 Volt liegen. Spitzen Ströme auf den **DATEN OUTPUTS** sollten auf ein Minimum reduziert werden.
*Der S4 ist für Stromspitzen normalerweise wie ein Scheinwiderstand, z.B. für Verbindungen zu **OUTPUT BUFFERN**. **EMUL**ation funktioniert nicht über Serienwiderstände mit einem hohen Scheinwiderstand.*

Emulieren von 2716

Aufgrund einer S4 Eigenheit ist es notwendig PROMs der Serie 2716 bei der Adresse 00800 zu emulieren.
Bei der Emulation muß der Zahlerncode zwischen 00800 und 00FFF stehen.
Wenn ein PROM der Serie 2716 geladen wird , kann beim LOAD - Befehl die Startadresse auf 00800 gesetzt werden. Dies wird durch drücken der ↑-Taste erreicht.

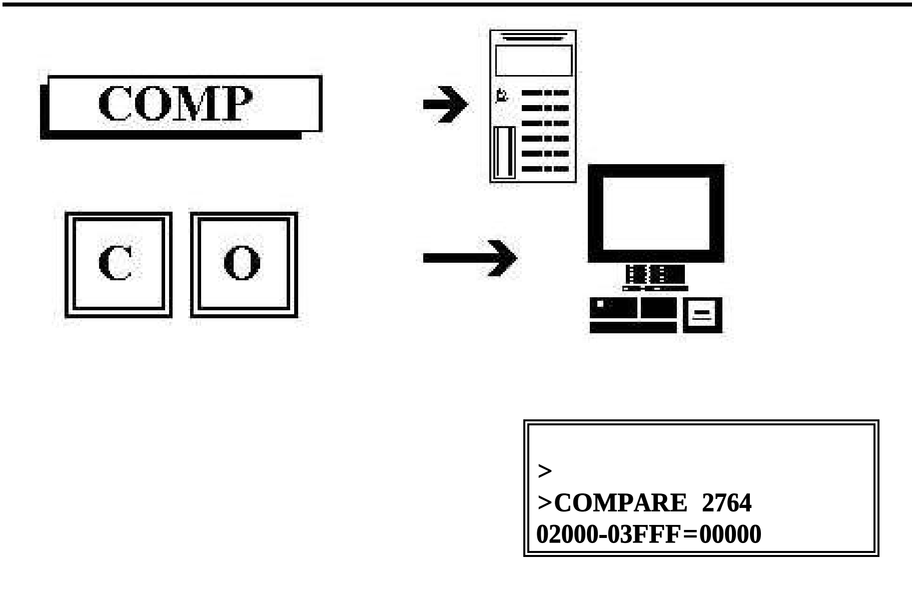
```
>LOAD  2716
00800-00FFF=00000
```

Falls die Daten im S4-Ram sind und auf ein PROM der Serie 2716 geladen werden soll, so ist es notwendig, diese auf die Adresse 00800 zu verschieben. Dabei dürfen sich der Start und Zielbereich nicht überschneiden.
Zum Beispiel:

```
>MOVE
00000,007FFF,00800
```

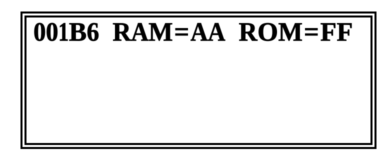
Verschiebt die Daten von 00000 bis 007FFF auf 00800.

COMP Taste



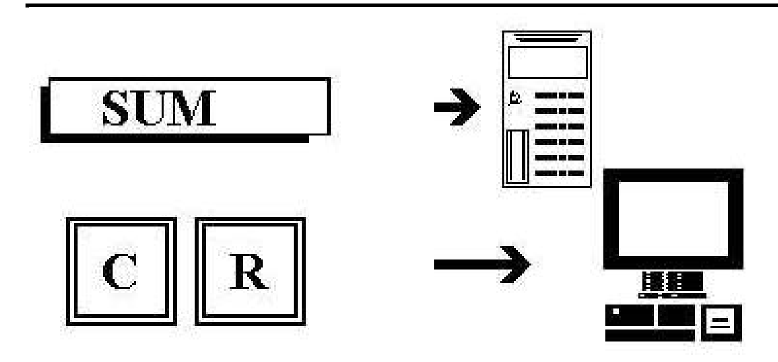
COMPARE (START) - (END) = (DEST)
START und **END** Adresse sind im **RAM**, die Bestimmungsadresse (**DEST**) ist im **PROM**. **COMP**are vergleicht alle Daten in einem **PROM** mit dem Inhalt, der im **USER RAM** zwischen der **START** und **END** Adresse liegt. Alle Daten, die nicht gleich sind, werden angezeigt. Wenn der Inhalt des **PROMS** genau dem des angegebenen Adressbereiches entspricht, wird **SAME** angezeigt.
Der S4 muß für den richtigen PROM Typ konfiguriert werden. Ein Warnhinweis gibt den PROM Typ an. Wenn ESCape während des Editierens der Parameter, oder während eines Reportes gedrückt wird, wird der Befehl abgebrochen. Um COMPare zu starten, oder um nach einem Reportfehler einen Neustart zu machen, drücken Sie die Enter Taste. Ein Fehler zwischen PROM und USER RAM wird angezeigt, und zwar wie folgt:

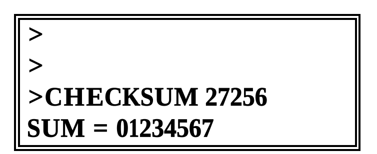
RAM ADRESSE   RAM BYTE   ROM BYTE



Um den nächsten Fehler anzuzeigen drücken Sie **ENTER**, zum Abbrechen **ESC**ape.

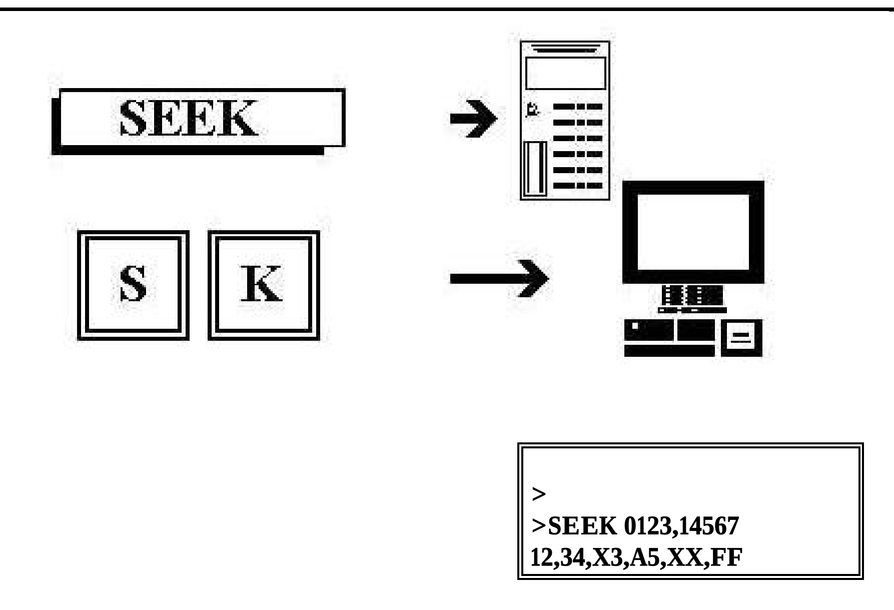
SUM Taste (grün)





Die grüne Taste **SUM** bearbeitet das **PROM** im **ZIF** Sockel. Es erstellt eine **CHECKSUM**me von dem **PROM** im **ZIF** Sockel. Die **CHECKSUM ROM** Funktion erlaubt es dem Benutzer die **CHECKSUM**me eines **PROM** Bausteins anzusehen, ohne ihn zuerst ins **USER RAM** zu laden. Dieses vermeidet das Überschreiben des **USER** Speichers. Der **PROM** Typ, der in der CONFIG ausgewählt wurde, wird angezeigt. Die **CHECKSUM** wird als 8-stellige Hexadezimalzahl dargestellt. Eine **CHECKSUM**me dient der Identifizierung von nicht beschrifteten **PROMS**. **MASTER PROMS** sollten mit ihrer **CHECKSUM**me beschriftet sein, damit sie nicht aus Versehen neu programmiert werden, und weder Bits verloren gehen, noch welche dazu kommen. Die **CHECKSUM** eines Bausteins, der falsch programmiert wurde, kann brauchbare Informationen liefern: Ob der Baustein mehr oder weniger Bits hat wie der **MASTER PROM**. Wenn die **CHECKSUM**me zu groß ist, könnte dies durch die Verwendung eines falschen Algorithmus oder durch einen Schaden entstanden sein. Ein **PROM**, daß als **MASTER PROM** verwendet wird, sollte mit seiner **CHEKSUM**me markiert werden. Die **CHECKSUM**me der einzelnen nachgebrannten **PROMS** sollte mit der des **MASTER PROMS** verglichen werden, um Fehler auszuschließen.

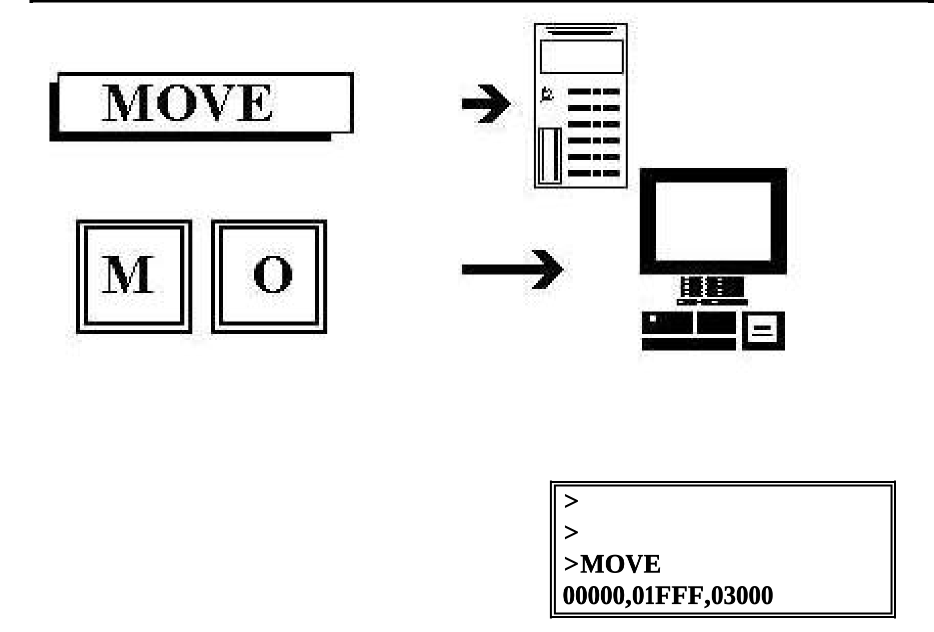
SEEK TASTE



>
>SEEK 0123,14567
12,34,X3,A5,XX,FF

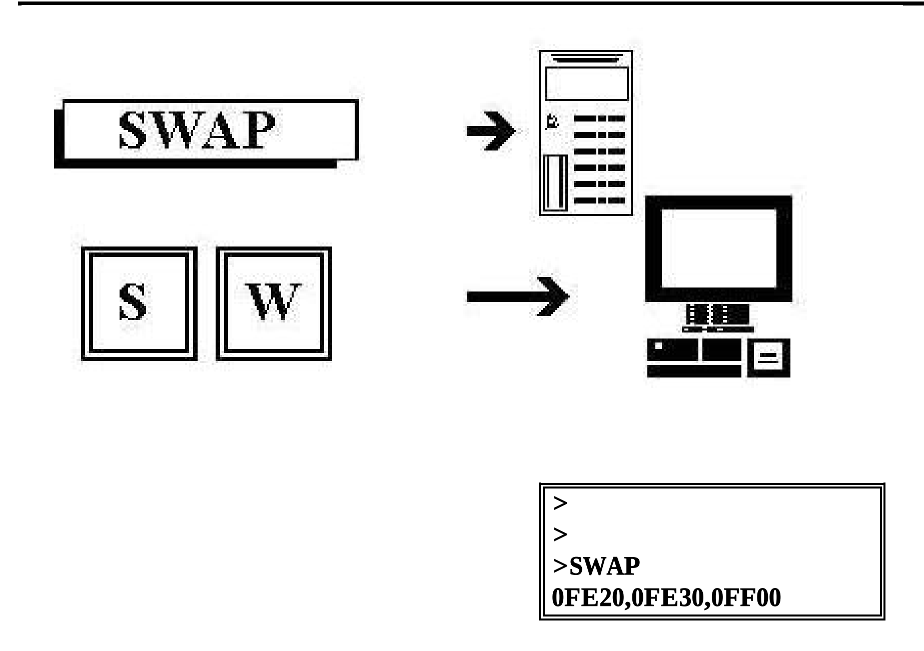
Um in einen Block von Codes, von einer (**START**) bis zu einer (**END**) Adresse, eine bestimmte Sequenz zu finden, die bis zu 6 Bytes haben kann. Bytes sind Paare von **HEX** Zeichen oder Platzhaltern, die jeden beliebigen Wert haben können. Platzhalter werden eingegeben, indem man die **SEEK** Taste noch einmal drückt (oder die X Taste am Terminal). Platzhalter werden mit **X** gekennzeichnet. Platzhalter können als halbe oder als ganze Bytes verwendet werden. Die erste Adresse, die gleiche Daten beinhaltet, wird im Display angezeigt. Drücken Sie **ENTER** um zur nächsten zu gelangen. Verwenden Sie **ESC**ape um die **SEEK** Funktion zu verlassen. Die **EDIT** Taste bringt Sie in den Editor, der an der angezeigten Stelle geladen wird. In allen Fällen läßt **SEEK** die zuletzt gefundene Adresse, in der **EDIT** Funktion, so daß Sie die gefundenen Daten editieren können.

MOVE Taste



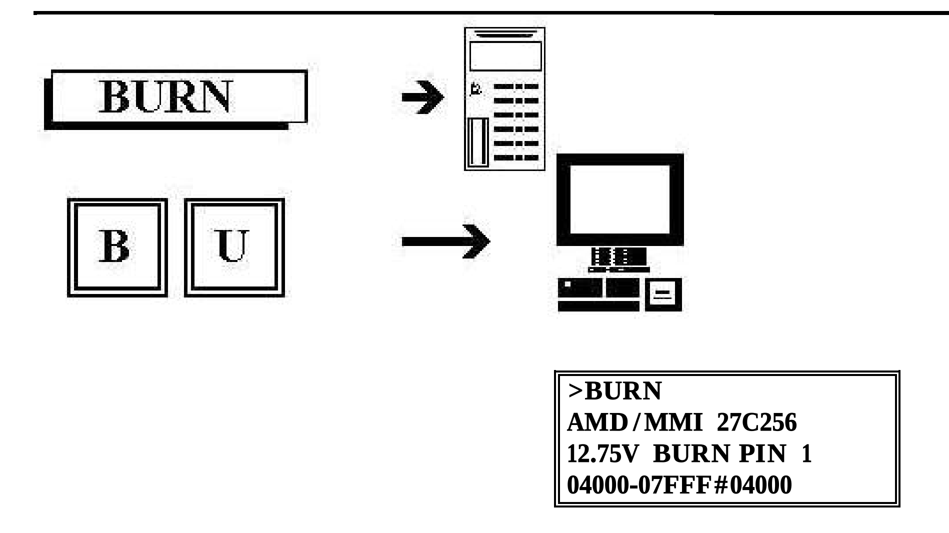
MOVE (START) , (END) , (DEST)
MOVE bewegt Speicher Blöcke im **USER RAM** , wobei **START** und **END** den Blockbereich angeben. Blöcke können vorwärts oder rückwärts bewegt werden. Überlappen verursacht keine Probleme. Die Inhalte des Originalblocks bleiben unverändert, außer wenn der Zielblock überlappt.

SWAP Taste



SWAP (START) , (END) , (DEST)
Tauscht den Code in zwei gleich großen Speicherblöcken.
Der Block von **HEX** Adresse (**START**) bis zu **HEX** Adresse (**END**) wird mit dem Block gleicher Länge, beginnend an der Adresse (**DEST**) getauscht. Das swappen von überlappenden Blöcken ergibt nicht viel Sinn, aber Sie können es trotzdem machen. Wenn Sie zwei mal hintereinander **SWAP**pen, dann ist der ursprüngliche Code wiederhergestellt, vorausgesetzt, daß die Adressbereiche nicht überlappt sind.

BURN Taste



BURN (START) - (END) = (DEST)
BURN programmiert das **PROM**, das zur Zeit im **ZIF** Sockel vorhanden ist, mit dem Inhalt aus dem USER **RAM**, der zwischen der **START** und **END** Adresse liegt.
Wenn der S4 wegen zu niedriger Batteriespannung automatisch abschaltet, während programmiert wird, ist das **PROM** trotzdem bis zu den Daten, wo der S4 dann abgeschalten hat, programmiert.
Der S4 muß für das richtige **PROM** konfiguriert werden. Ein Warnhinweis gibt den **PROM** Typ, die Programmierspannung und den Pin, an den die Spannung gelegt wird, an. Wenn eine dieser Einstellungen falsch ist, kann das **PROM** zerstört werden. Vergessen Sie nicht, daß die Pinbelegung der 24 oder 28 Pin **PROMS** unterschiedlich zu der von 32 Pin **PROMS** ist.
Wenn die **ESC**ape Taste während des Editierens der Parameter oder nach dem Programmstart gedrückt wird, wird das Programm abgebrochen.
Um den **BURN** Zyklus zu starten, drücken Sie **ENTER**. Wenn der Vorgang abgeschlossen ist, dann geht das Programm in den **COMPARE** Modus um die Inhalte zu Vergleichen.
Folgendes Übungsbeispiel: Programmieren Sie ein 27256 **PROM** im **ZIF** Sockel, von der Adresse **04000** bis **05FFF** mit dem Inhalt des **USER RAMS** von der Adresse **04000** bis **05FFF**. Der Rest des **PROM** soll während des Programmieren und dem Testen ignoriert werden.

Fehler beim Programmieren

Falls beim brennen des EPROMs ein Fehler auftritt, erhalten Sie folgende Meldung:

```
FAILED  AT  00000
00000  RAM=3E  ROM=FF
```

Der S4 zeigt die Adresse an, bei der der Fehler aufgetreten ist, zeigt aber auch an welche Daten im RAM und welche im ROM stehen. Um den nächsten Fehler anzuzeigen drücken Sie **ENTER**, um abzubrechen **ESC**ape.
Folgende Punkte sollten beachtet werden:

1. Der S4 muß auf den richtigen Prom Typ eingestellt werden. Mit der **PROM** - Taste können Sie diesen auswählen. ***Eine falsche Einstellung kann zur Beschädigung des EPROMs führen!***
2. Versichern Sie sich, daß das EPROM komplett gelöscht (blank) ist. Der BLANK - Test wird mit der **TEST** - Taste gemacht.
3. Wenn Sie 24- oder 28-pin EPROMS verwenden, so sind diese am unteren Ende des ZIF-Sockels mit der Einkerbung nach oben zu positionieren.

Programmieren von 2716s

Aufgrund einer Eigenheit der S4 Hardware müssen EPROMS der Serie 2716 auf eine spezielle Art programmiert werden.
Normalerweise ist die Startadresse beim programmieren im RAM 00000. **Beim 2716 muß diese 00800 sein!**
Diese Adresse zu ändern ist sehr einfach:
Die S4-Funktionen **LOAD, BURN, TEST** und **COMP** verwenden Adressen in diesem Format:

XXXXX,YYYYY = ZZZZZ

XXXXX ist die Startadresse
YYYYY ist die Endadresse
ZZZZZ ist die Startadresse des EPROMs

Wir müssen XXXXX ändern, und dazu verwenden wir einen der Befehle **LOAD, BURN, TEST** oder **COMP:** z.B.:

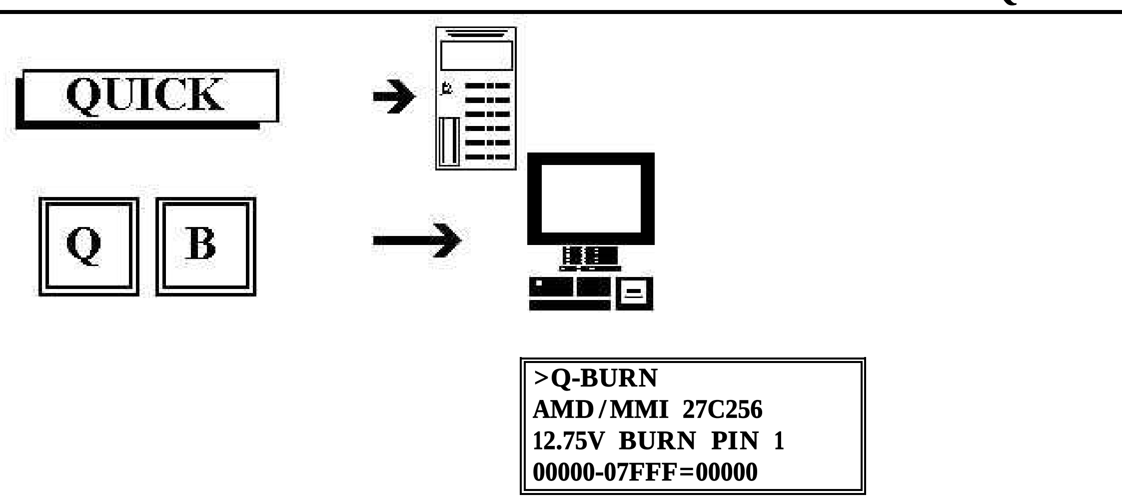
```
>LOAD  2716
00000,007FF=00000
```

Der Curser blinkt beim ersten Digit der Startadresse. Mit drücken der ↑- Taste ändert sich die Start- und Zieladresse wie folgt:

```
>LOAD  2716
00800,00FFF=00000
```

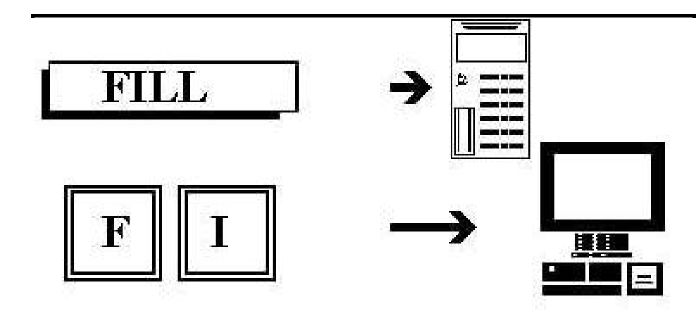
Bei EPROMs der Serie 2716 muß immer mit diesen Adressen gearbeitet werden, da sonst eine Richtige Funktion nicht garantiert werden kann!

QUICK Taste



Q - BURN (START) - (END) = (DEST)
QUICK brennt ein **PROM** viel schneller, wenn nur ein paar Locations benötigt werden. Jede Adresse wird überprüft und übersprungen wenn die Daten gleich sind. **EEPROMS** werden nicht gelöscht. **QUICK** ist nicht immer schneller. Wenn das **PROM** z.B. komplett programmiert werden muß, oder wenn sehr viel unterschiedlich ist, ist das normale **BURN** oft schneller. **PROMS** die normalerweise in **QUICK PULSE** oder **FLASH RITE** Algorithmen programmiert werden, zeigen keinen Unterschied in der **BURN** Zeit an. Weil der Vergleichszyklus der **QUICK BURN** genau so lange dauert, wie das Programmieren bei **BURN**.
*Wie auch immer, um ein paar Bytes in einem **PROM** zu ändern, was normalerweise ein paar Minuten dauert, kann **QUICK** Zeit sparen.*

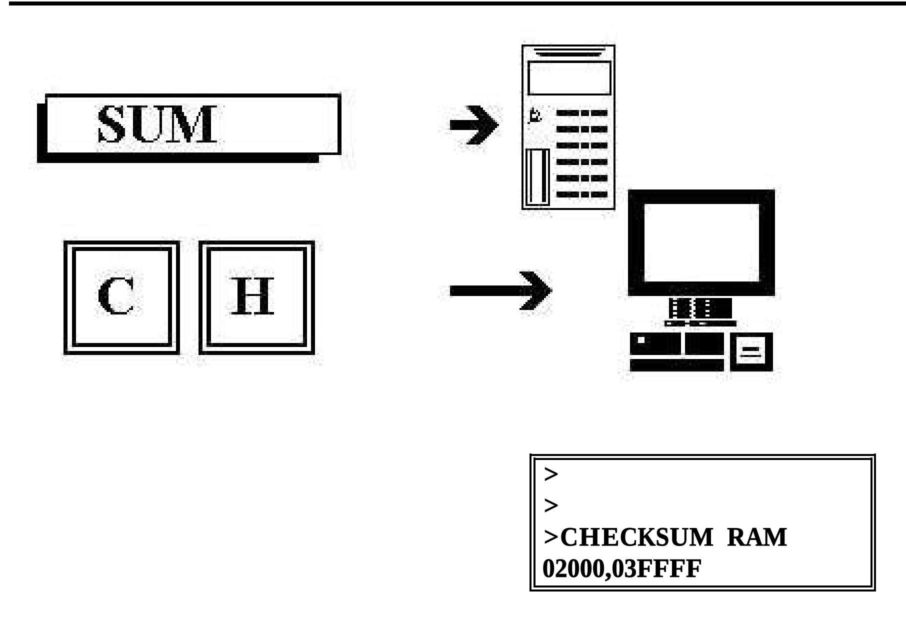
FILL Taste





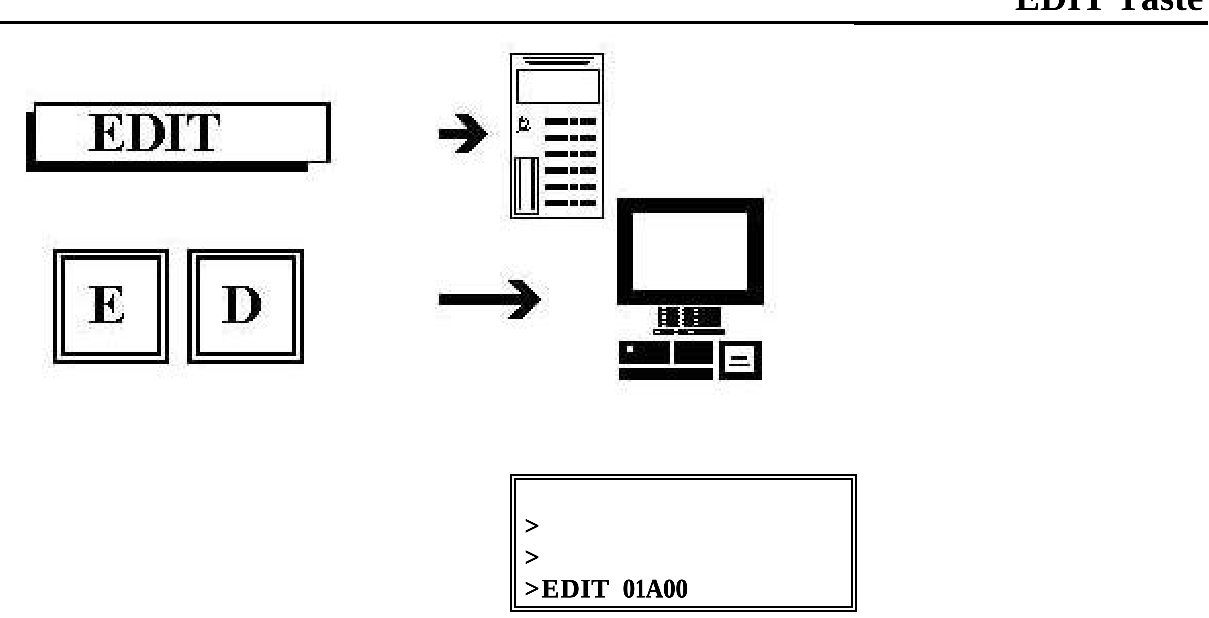
FILL (START) , (END) (BYTE)
Füllt jedes Byte im Speicher mit einem Hex Wert (**BYTE**) zwischen den Hex Adressen (**START**) und (**END**) {inklusive beider Adressen}.

SUM Taste (grau)



CHECKSUM (START) , (END)
Die **graue** Taste **SUM** addiert alle Bytesvom RAM in dem ausgewählten Block zwischen der **START** und der **END** Adresse (inklusive beider Adressen). Die Funktion präsentiert das Ergebnis als 8-stelligen Hex Code.
Die **grüne SUM** Taste hat eine ähnliche Funktion, allerdings bei **PROMS**.

EDIT Taste



EDIT ist in zwei verschiedene Funktionen unterteilt, um das jeweils Beste aus der **LCD** Anzeige oder dem Terminal zu machen.

Stand-alone Editieren

Der S4 verwendet das ganze **LCD** Display während der **EDIT** Routine.

ADR = 01A00	**ABCD**	**EFGH**	
4142	**4344**	**4546**	**4748**
494A	**4B4C**	**4D4E**	**4F50**
5152	**5354**	**5556**	**5758**

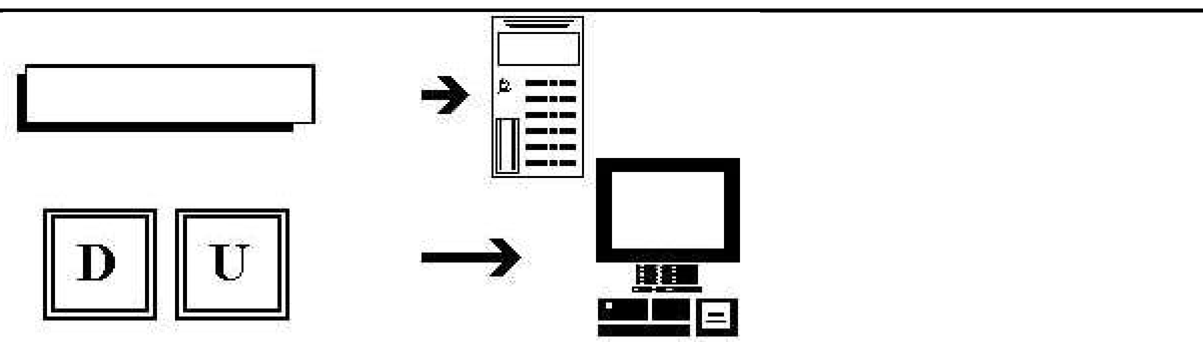
Die oberste Zeile zeigt die aktuelle Cursor Position an, die geändert wird, wenn Sie eine der Cursortasten drücken, oder wenn Sie Daten eingeben. Am rechten Ende der ersten Zeile ist die derzeitige Zeile im ASCII Code. (Um den ASCII Code zu übersetzten, wird Bit 7 ignoriert). Werte zwischen 20H und 7FH werden als
• (Punkt) dargestellt. Der Cursor erscheint als blinkender Block (⊂), der jedoch zu einem unterstrichenem Cursor (_) wird, wenn Sie eine Cursortaste gedrückt halten, um sich schnell durch den Speicher zu bewegen. Wenn Sie am oberen oder am unteren Rand der Anzeige angekommen sind, dann scrollt die Anzeige nur noch um jeweils eine Zeile. Editieren ist nicht ungefährlich, denn wenn ein Code geändert wurde, gibt es keine Möglichkeit den Originalcode wiederherzustellen. Die **ENTER** Taste kann verwendet werden um die Adresse zu ändern, ohne **EDIT** verlassen zu müssen.

Terminal Editieren

Der Terminal Bildschirm hat mehr Spalten als eine LCD Anzeige und schaut ungefähr so aus:
00000 42 43 44 45 46 47 48 49
Der **ADRESSE** folgen die Daten in acht Speicherbereichen. Wie üblich bei Terminaleingaben, wird **SPACE/BACKSPACE** verwendet um sich durch den Speicher zu bewegen. Mit den Tasten **0-9** und **A-F** ändert man den Hex Code der aktuellen Adresse. Die **ADRESSE** kann geändert werden, indem man mit **BACKSPACE** in das Adressenfeld fährt.
Wenn die **ENTER** Taste gedrückt wird, dann wird der gleichwertige **ASCII** Code dieser Zeile am Ende angezeigt, und die nächsten acht Byte werden zum editieren angezeigt:
00000 42 43 44 45 46 47 48 49 BCDEFGHI
00008 00 B1 46 92 8F FF 91 0C
Nur **ASCII** Zeichen von 20H bis 7EH werden über die Schnittstelle übertragen, nachdem das **MSB** vollständig durchsucht wurde. Die anderen Zeichen werden als Punkt (**FULL STOP**) übermittelt, weil Bedienungsparameter vom Terminal als Befehl interpretiert, und nicht auf den Bildschirm geschrieben werden würden. Um aus der **EDIT** Routine auszusteigen, drücken Sie die **ESC**ape Taste.

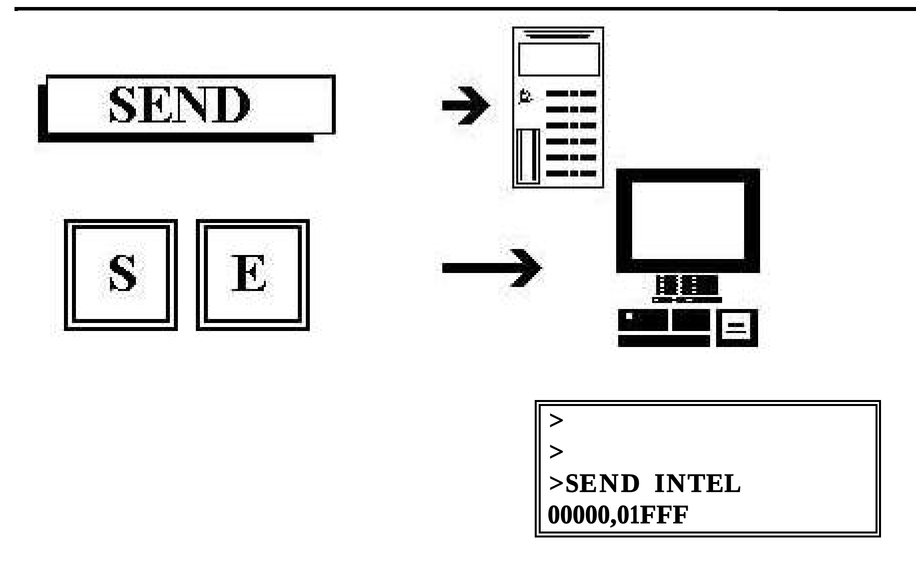
DUMP RAM





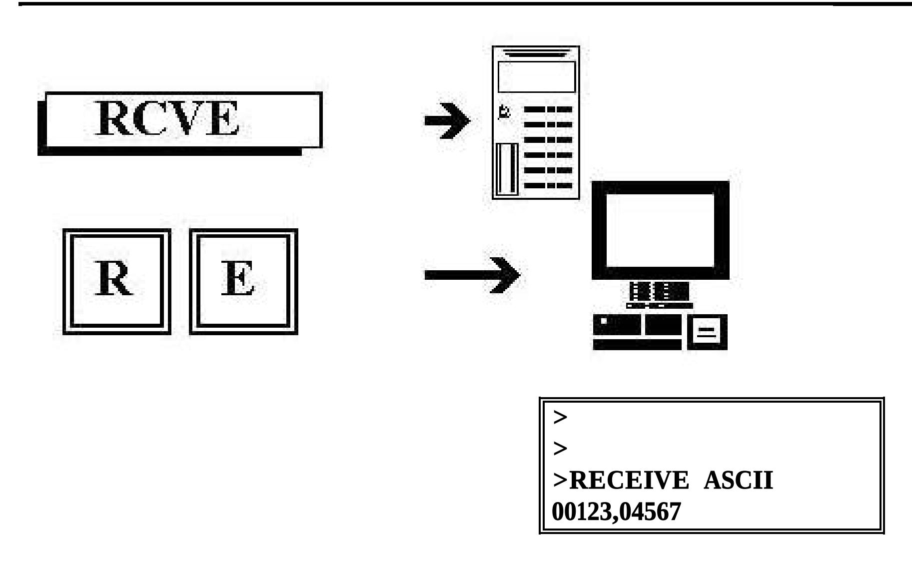
DUMP (START) , (BYTES)
DUMP arbeitet nur mit dem Terminal und ist gleichwertig mit **EDIT**. Es verändert 8 Bytes pro Zeile im **HEX** Code mit gleichwertigem **ASCII** Code. Eingaben die verlangt werden sind **START** Adresse und die Anzahl von **BYTES** bis zu **FFF** oder **255**. Wenn 0 Bytes eingegeben werden, wiederholt **DUMP** diese Abfrage, bis **ESC**ape gedrückt wird.

SEND Taste



SEND (START) , (END)
Übermittelt den Block von der **START**, bis inklusive der **END** Adresse über die serielle Schnittstelle. Das **FILE FORMAT** und die **BAUD RATE** können im **SETUP** umgestellt werden. Wenn **HANDSHAKE** auf **RTS** oder **DTR** gesetzt ist, dann muß dieses Signal auf „ **true**“ sein, bevor die Übertragung beginnt. Die Übertragung funktioniert dann nur, wenn **RTS** „**true**“ ist, wenn **RTS** „**false**“ wird, dann stoppt die Übertragung so lange, bis **RTS** wieder „**true**“ wird. Die Übertragung kann an jeder Stelle mit **ESC**ape abgebrochen werden.

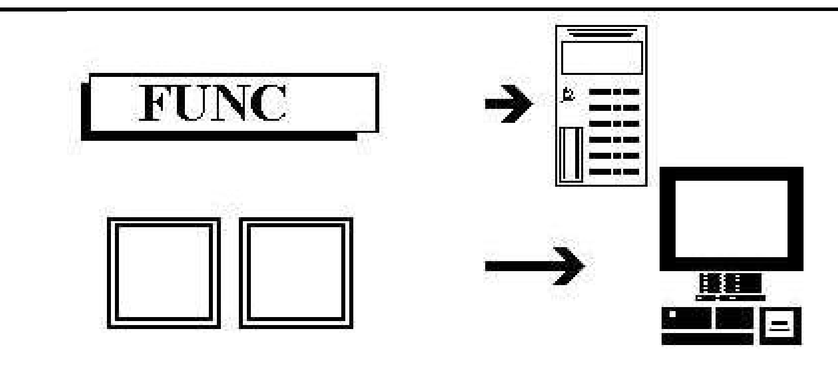
RCVE Taste



Der S4 empfängt Files im **INTEL**, **MOTOROLA**, **TEKHEX**, **ASCII** oder **BINARY** Format, wie im **SETUP** definiert. Sie müssen die **START** und **END** Adresse, nur möglich im **ASCII** oder **BINARY** Format, angeben, weil diese Übertragung keinen Zielort (**DEST**) für die Daten enthält. Wenn die Übertragung fortsetzt, nachdem die **END** Adresse erreicht ist, dann läuft das File aus, z.B. Daten werden empfangen, aber nicht abgelegt.
Die **ESC**ape Taste am Keypad bricht Übertragungen ab. Sie bricht auch während der Annahme von Files ab. Nachdem die **ESC**ape Taste gedrückt wurde, wartet der S4 für ein paar Sekunden, um sicher zu stellen, daß keine Daten mehr kommen. Wenn doch, würden solche Daten von S4 als Keyboard Funktionen mißverstanden werden. Auch der sendende Terminal überprüft den **RTS**, ob dieser „**true**“ bleibt, bis das **END OF FILE** erreicht ist. Wenn noch mehr Daten kommen, bleibt der S4 in der Routine. In so einem Fall sollte die Übertragung am sendenden Ende abgebrochen werden. **ERROR CHECK** Fehler im File Format werden angezeigt - z.B. **CHECKSUM** Fehler werden angezeigt. Wenn ein File korrekt empfangen wurde, zeigt S4 die letzte Adresse an, wo Daten gespeichert wurden.

Bevor der S4 in den Befehls Modus zurückkehrt, wartet er bis der Dateninput für in paar Sekunden inaktiv ist. Dies geschieht, um zu vermeiden, das irgendwelche Extradaten nicht mehr aufgenommen werden und dadurch das Terminal abstürzt. Wenn ein Batch File gesendet wird, muß eine Verzögerung von mindestens 1 Sekunde zwischen den Kommandos sein, um zu vermeiden, daß der S4 es als Extradaten interpretiert und nicht speichert.

FUNC Taste

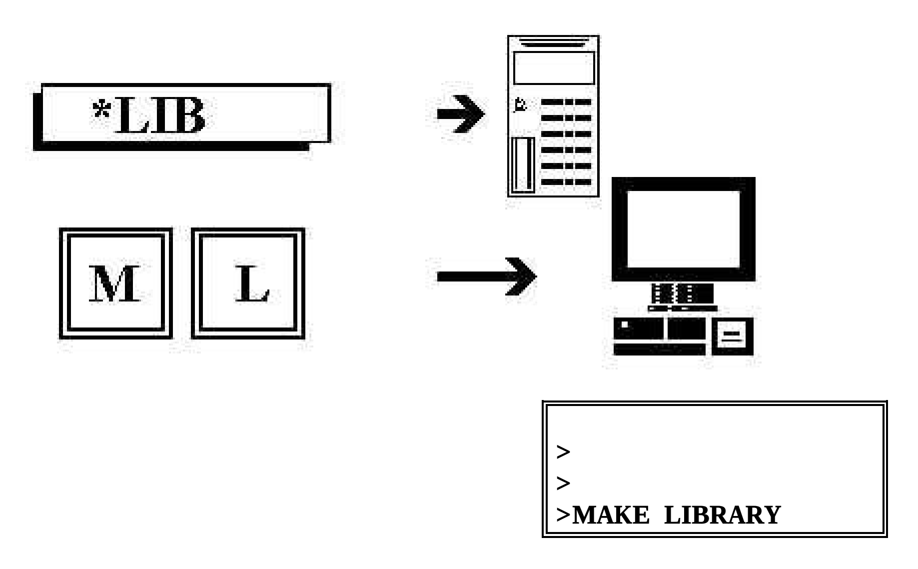


Die FUNK Taste arbeitet wie die SHIFT Taste, sie ermöglicht den Zugriff auf Zweitfunktionen der Keypad Funktionen. Es gibt keine gleichwertige Tastenkombination auf dem Terminal, da sowieso jede Funktion ihren eigenen, zweistelligen Tasten Code hat.



Um anzuzeigen, daß eine Zweitfunktion ausgewählt wurde, ändert sich die Anzeige von einem > in ein *.

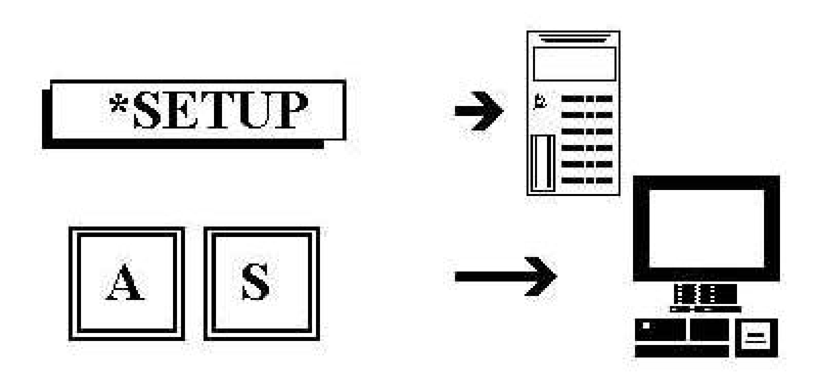
FUNC LIB = Erstellen eines Library



MAKE LIBRARY ist das Gegenteil vom Laden eines Librarys. Es erstellt ein **LIBRARY** PROM von dem Code und den Einstellungen, die zur Zeit im **TPA** vorhanden sind. **MAKE LIBRARY** gibt Ihnen die Möglichkeit, ein **LIBRARY PROM** mit Ihren eigenen Einstellungen zu erstellen. Der Code aus dem **TPA** wird in das **USER RAM** ab der Adresse 8000 aufwärts kopiert. Dabei kann eine Aufspaltung der Daten kommen (max. 42 Bytes - genaueres im **LIBRARY**). Alles was Sie machen müssen ist, nachdem **MAKE LIBRARY** für ein 27256er konfiguriert wurde, es mit dem **USER RAM** von 08000 bis 0FFFF zu programmieren. Ein solches **LIBRARY ROM** speichert den **PROM ALGORITHMUS**, die **BAUD RATE**, den **FILE TYPE** und andere System Variablen, die gesetzt waren. Wenn Sie Ihren S4 einem Kollegen borgen, ist es ratsam ein **LIBRARY**

ROM mit Ihren Daten anzufertigen. Denn wenn Sie ihn zurück bekommen, müssen Sie nur das **LIBRARY** neu laden, und alle Einstellungen sind wieder so, wie Sie es gewohnt sind.

FUNC SETUP = Advanced



>ADVANCED SETUP	
SHUTDOWN TIME	1E
HIGH TONE	98
LOW TONE	AC
BUSY TONE	50
MAX BATT TEMP	AA
MIN BATT TEMP	64
CHARGE TIME	0B
DISCHARGE TIME	1C
DEEP DISCHARGE	40
NORM DISCHARGE	C0

Um im Advanced Setup die Parameter ändern zu können, müssen Sie die Nummern von Dezimal ins Hexadezimale übertragen.

Shutdown Time (Abschaltzeit) 1E = 30 Minuten

Die hörbaren Töne, die der S4 machen kann, können untereinander ausgetauscht werden, oder mit 00 ganz abgeschalten werden.

High Tone (Hoher Ton) 98 = 1520 Hz
Low Tone (Tiefer Ton) AC = 1720 Hz
Busy Tone (Schneller Ton) 50 = 800 Hz

Das Ladegerät wird vom Laden abgehalten wenn die Akkus zu heiß oder zu kalt sind (Min. und Max). Wenn Sie diese Grenzen verändern wollen, 1°C entspricht ungefähr 2,7.

Max Batt Temp AA = 42°C
Min Batt Temp 64 = 15°C

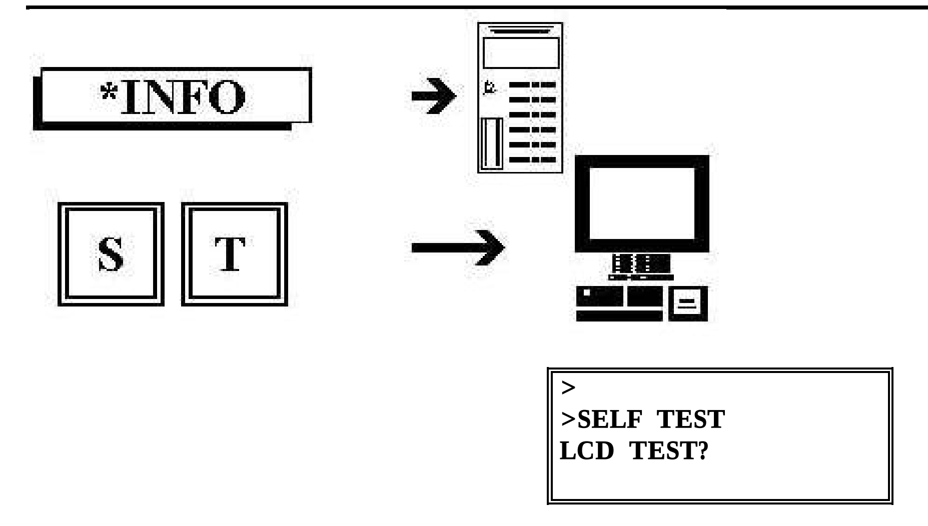
Die folgenden Parameter verändern die Operationen des Ladungszustandsmelder. Ladezeit und Entladezeit sind proportional zu den Sekunden, die gebraucht werden um die Akkuladung um eine mAh zu laden. Kleinere Werte machen die Lade- und Entladezeiten schneller. Die Werte können verändert werden, wenn ein genauerer Ladungszustandsmelder nötig ist.

Charge Time (Lade Zeit) 0B
Discharge Time (Entlade Zeit) 1C

Deep und Normale Entladung sind automatische Operationen des Laders. Diese sind auf 25% und 75%, bei einem Kaltstart gesetzt. Normalerweise wird der Akku geladen, wenn das Ladegerät angesteckt ,der S4 eingeschaltet ist und der Ladungszustandsmelder weniger als das Norm Discharge Level anzeigt. Wenn der S4 benutzt wird, und das Ladegerät angesteckt ist, kann der Ladungszustandsmelder unter das Deep Discharge Level fallen, bevor die Batterie wieder geladen wird.

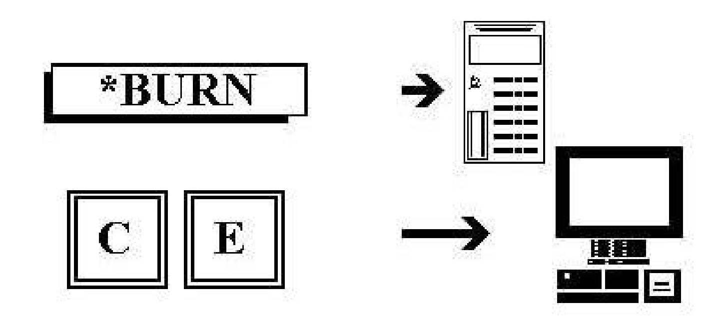
Deep Discharge 40
Norm Discharge C0

FUNC INFO = Selbsttest



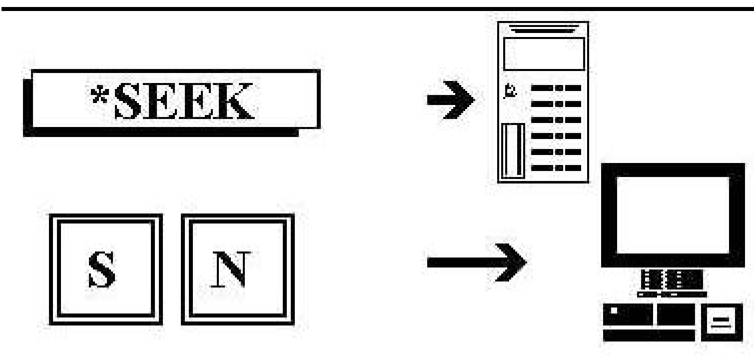
Dieser Test wird verwendet, um zu überprüfen, ob die **LCD** Anzeige, das Keypad und das **USER RAM** richtig funktionieren. Mit **ENTER** starten Sie diesen Test. Mit **ESC**ape gehen Sie über zum nächsten Test.
Der **LCD TEST** füllt das Display mit einer Zeichen Sequenz.
Beim **KEYPAD TEST** müssen Sie die Tasten, von Links nach Rechts und von Oben nach Unten drücken. Der Test kann nicht abgebrochen werden, außer wenn alle Tasten gedrückt wurden.
Der **DESTUCTIVE RAM TEST** füllt das **USER RAM** mit einer Zufalls Nummern Sequenz, und vergleicht diese mit der gleichen Sequenz. ***Der originale Speicherinhalt geht dabei verloren!***

FUNC BURN = Chip löschen



Dieses Kommando kann alle **FLASH PROMS** löschen.
FLASH PROMS können neu programmiert werden, dabei ist es notwendig diese zuerst zu löschen. Chip löschen ist mit neuen und gelöschten **PROMS** nicht notwendig und automatisches Löschen verlangsamt nur das Programmieren.

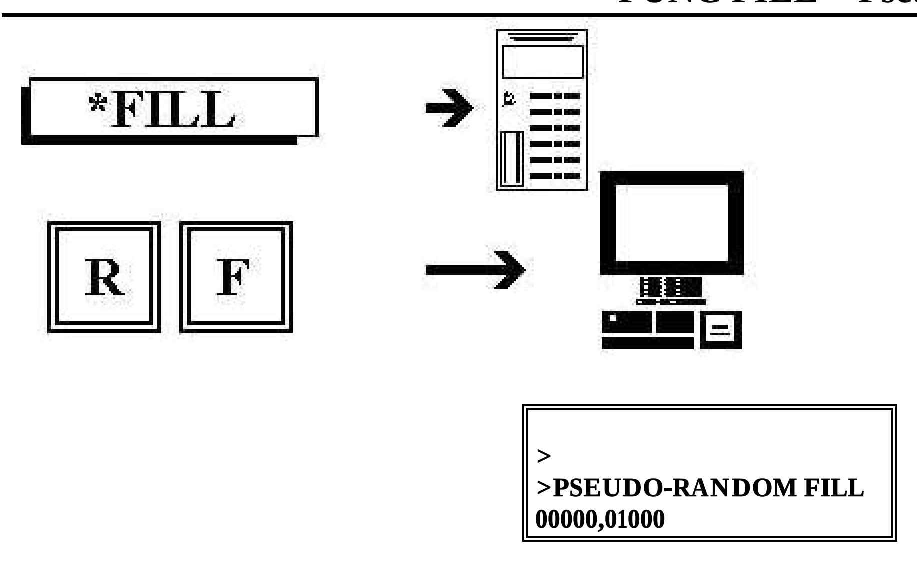
FUNC SEEK = Seek not Equal to



>
>**SEEK NOT EQUAL TO**
00000,1FFFF,FF

Dieses Kommando ist das Gegenteil zu **SEEK**. Es findet alle Bereiche, die nicht mit dem definiertem Byte übereinstimmen. Es kann nützlich sein, um **PROM** Bereiche zu finden, die nicht gelöscht wurden.

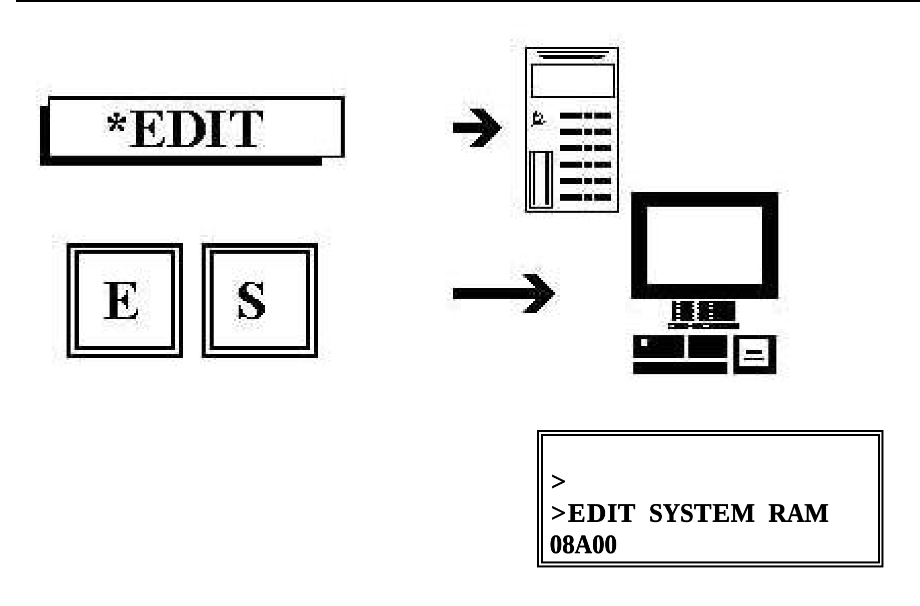
FUNC FILL = Pseudo Random FILL



>
>**PSEUDO-RANDOM FILL**
00000,01000

Füllt den Speicher mit Pseudo Random Nummer Sequenzen von der **START** bis inklusive **END** Adresse. Pseudo Random Nummern werden von einem Programm erzeugt. Diese sind keine wirklichen Zufallszahlen. Wenn Sie welche sein würden, dann würden sie immer unterschiedlich sein, und man könnte sie nie vorhersagen, was man aber kann. Es sind daher immer wiederkehrende Sequenz, die aber für diesen Test zufällig genug sind. Der Algorithmus ist wie folgt: Ein 15 - Bit linksschieben, mit dem letzten signifikanten Bit wird eine **XOR** Verknüpfung mit den Bits 0 bis 14 vorgenommen. Diese Sequenz wiederholt sich alle 32,767 Linksschiebungen. Die niederen 8 Bits werden verwendet. Jedes Byte schaut aus wie eine Linksverschiebung des aktuellen Bytes aus, außer für den LSB, der nicht vorhersehbar ist.

FUNC EDIT = Editieren des System RAM



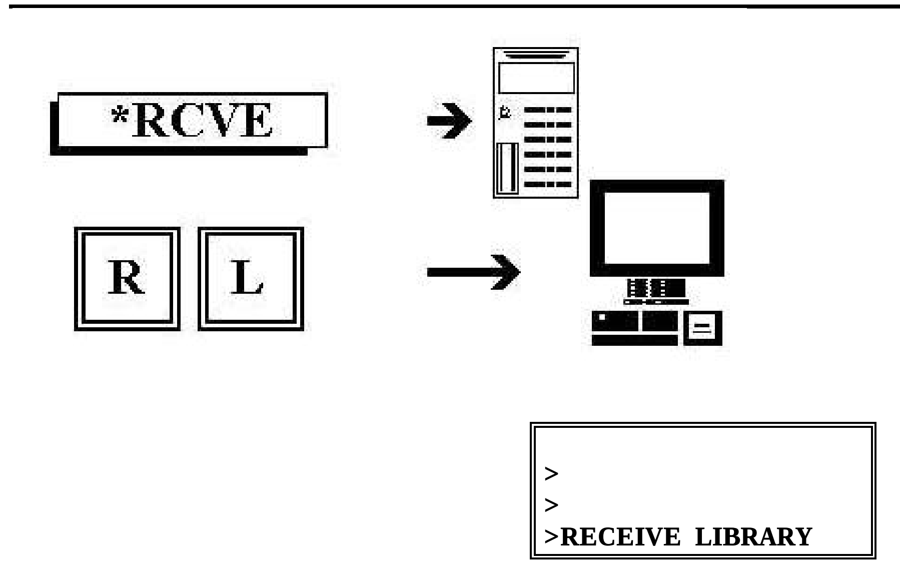
>
>**EDIT SYSTEM RAM**
08A00

Der System Speicher kann mit einer **FUNC** Version von **EDIT** angesehen und verändert werden. System Speicher ist der Bereich, in den Programme geladen werden, nachdem Sie von einem **LIBRARY ROM** in das Transient Program Area von **8000** bis **FFFF** geschrieben wurden. Wenn die **FUNC** Taste gedrückt wurde, ändert sich das **>** in ein *****. Wenn jetzt **EDIT** gedrückt wird, arbeiten Sie im Systemspeicher. Die Anzeige verhält sich genau wie die normale **EDIT** Routine.

Warnung *EDIT ermöglicht das Ändern von Arbeitsprogrammen und System Variablen. Der S4 kann abstürzen, wenn das Startprogramm falsch geändert wurde, dies kann auch beim Ändern des Stack Codes passieren. Betrachten Sie das TPA als „NO GO“ Area, außer Sie wissen genau was Sie wollen. Sie können alle Änderungen sofort mit RESET abbrechen, und anschl. das LIBARY ROM wieder laden.

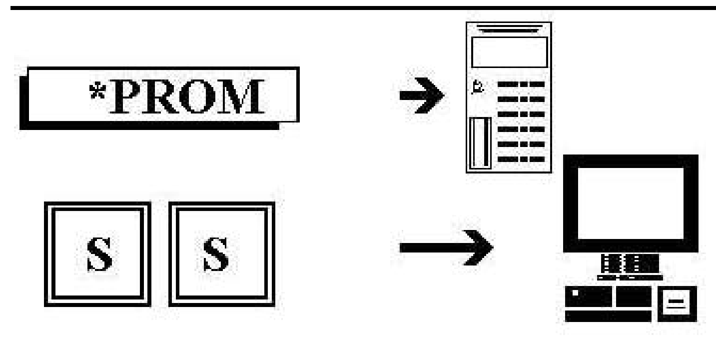
```
ADR = 0E00    DATA  MAN
4441    5441  4D41   4E20
53B3    0838   3838   0C06
0129    1267  0008   85F9.
```

FUNC RCVE = Receive Library



Überspielt ein neues **LIBRARY ROM** in den **TPA**. Dieses Kommando ist zu verwenden, wenn Sie ein neues Library vom Terminal laden wollen. Wenn Sie das File zum S4 geschickt haben, können Sie mit **MAKELIB** ein neues **LIBRARY ROM** erzeugen. Das Setup muß dazu auf **INTEL** Files gestellt sein, um richtig zu arbeiten. Wenn das Library empfangen wurde, werden Sie gefragt, ob Sie dieses starten wollen. Wenn Sie starten wollen, drücken Sie **ENTER**, wenn nicht drücken Sie **ESC**ape.

FUNC PROM = Silicon Signature



Silicon Signaturers werden für die Identifikation der Bauteile verwendet. Die meisten modernen PROMs verwenden diese Kennzeichnungsmethode. Mit deren Hilfe ist es dem S4 möglich sich selbst zu konfigurieren.
Vorgangsweise:
Stecken Sie den Prom in den ZIF-Sockel und drücken Sie **FUNC-PROM** oder **SS** bei der Tastatur.Der S4 liest dann die Silicon Signature aus und erhält damit den Herrstellercode und den PROM-Code. Er vergleicht diese beiden Informationen mit dem Library. Wenn er eine exakte Übereinstimmung findet, zeigt der S4 den Hersteller und den PROM Typ an und konfiguriert sich selbst.
Mit **ENTER** oder **ESC**ape verlassen Sie die *PROM-Option.

Falls das PROM keine Silicon Signature enthält erhalten Sie folgende Meldung:

```
>
>SILICON SIGNATURE
NO SIGNATURE FOUND
```

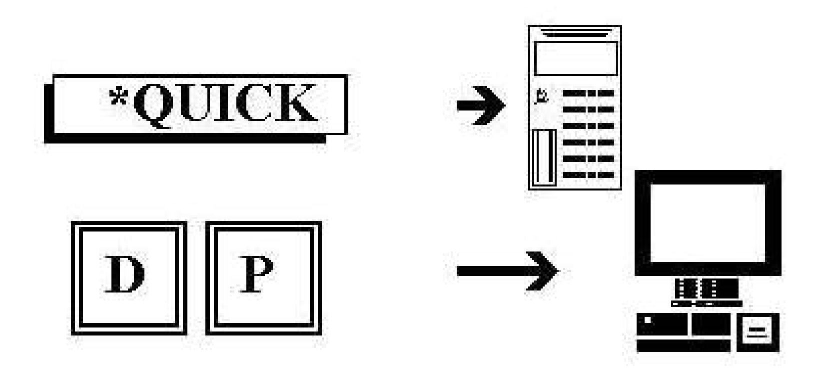
Bauteile die keine Silicon Signature besitzen, können bei dieser Abfrage beschädigt werden wenn diese auch keinen internen Überspannungsschutz besitzen. Da bei der Abfrage die Spannung an den Adressleitungen bis zu 12V angehoben wird.

Der S4 hat nicht alle Codes der Silicon Signature gespeichert. Falls er ein PROM nicht zuordnen kann erscheint folgende Meldung:

```
>SILICON SIGNATURE
SIGNATURE: 8F23
NO MATCH FOUND
>
```

In diesem Fall kontaktieren Sie bitte unseren Technischen Support. Wir werden diese Produkte in unser nächstes Library-Update mit aufnehmen.

FUNC QUICK = Data Protection



Ein Softwareschutz ist ein Feature von modernen EEPROMs und 5V Flash-ROMs. Ist dieser Schutz aktiviert ist es nicht möglich Daten zu lesen oder das Bauteil zu programmieren. Beim Softwareschutz wird eine bestimmte Sequence von Bytes auf eine bestimmte Adresse im ROM geschrieben.

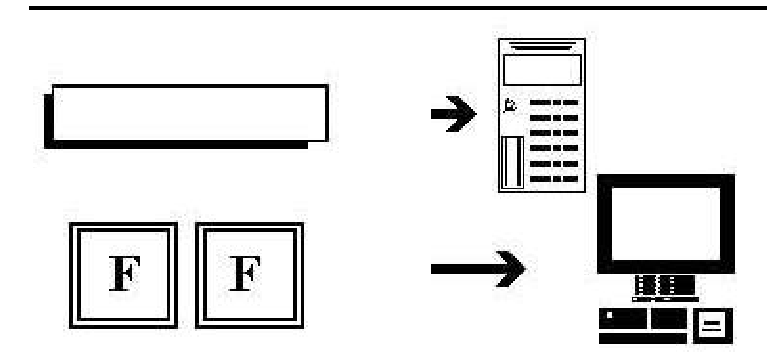
Bevor Sie den Softwareschutz ein- oder ausschalten, vergewissern Sie sich ob ihr Bauteil diese Option auch unterstützt. Stecken Sie dann das Bauteil in den ZIF-Sockel und drücken Sie **FUNC-QUICK** oder **DP** auf der Tastatur.

```
>DATA PROTECTION
DATA PROTECT OFF
```

Verwenden Sie die ← oder → Taste um den Schutz ein- oder auszuschalten. Mit **ENTER** verlassen Sie diese Option.

Bem: Die 5V Flash ROMs AMD 29FXXX haben einen ständigen Softwareschutz, der auch von dieser Funktion nicht ausgeschaltet werden kann.

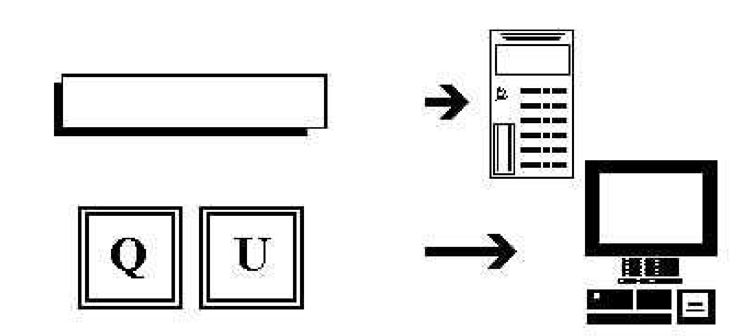
FILE FORMAT



>FILE FORMAT
INTEL

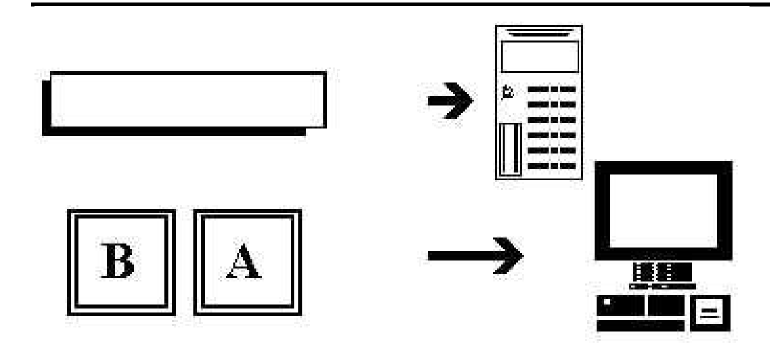
Hat keine gleichwertige Keypad Funktion. Die Leertaste und die Backspace Taste ändern das File Format von **ASCII INTEL MOTOROLA TEKHEX BINARY**. Entweder mit **ESC**ape oder **ENTER** verlassen Sie die Funktion, das zuletzt angezeigte Format wird jetzt am Bildschirm angezeigt.

QUIET MODE



Hat keine gleichwertige Keypad Funktion. **QUIET MODE** schaltet den **RS232** Ausgang ab. Es gibt keine Anzeige am Bildschirm, bis das Kommando eingegeben wurde. Diese Funktion wird verwendet bei Batch Programmen die direkt auf **COM1** schreiben etc..

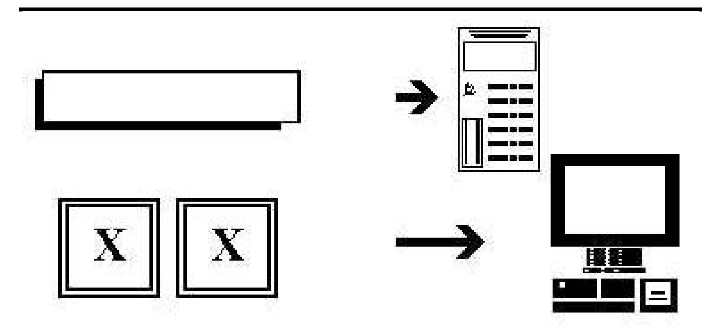
BAUD RATE



>SET SERIAL PORT
9600
Hat keine gleichwertige Keypad Funktion. Die Leertaste und die Backspace Taste werden verwendet um die Auswahl zu zeigen:
300, 600, 1200, 2400, 4800, 9600, 14400, 28800, 115200

Mit ESCape oder **ENTER** verlassen Sie das **BAUD RATEN** Setup beim zuletzt angezeigten Wert.
Beim Verlassen müssen Sie die Terminal Baud Rate eingeben.

EXECUTE



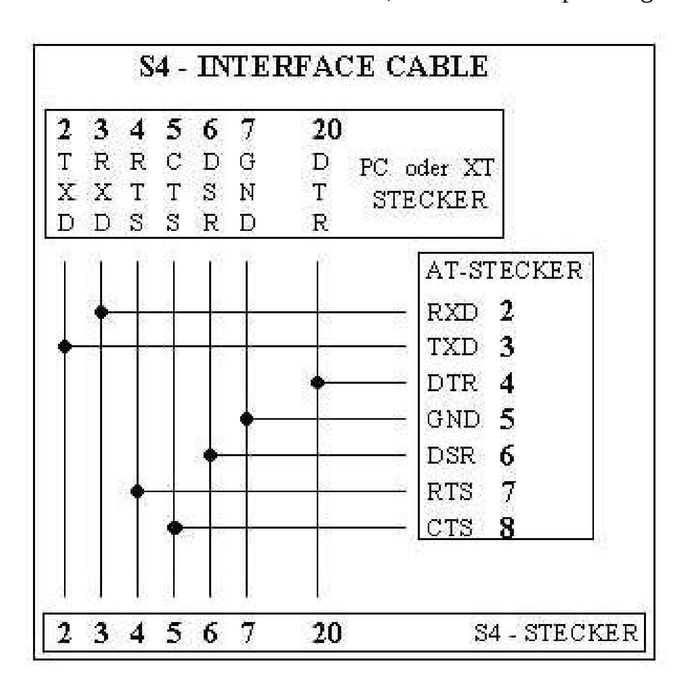
>EXEC 01234

Startet Programme ab der angegebenen Adresse. Das ist hauptsächlich für Programmierer, die neue Programme für den S4 schreiben. Die Vorgabeadresse ist die zuletzt in der **EDIT** Funktion verwendete Adresse. Sie können jede Adresse setzten, die Sie wollen.

00000 verursacht einen Kaltstart
0F000 verursacht einen Warmstart

RS232 Serielle Schnittstelle

Die serielle RS232 Schnittstelle verwendet den MOTOROLA-IC MC145406, welcher alle Spannungen und Ströme sowie Ein und Ausgänge treibt.
Der Strom kann von irgendwelchen Ausgängen geliefert werden, darf aber 5mA nicht überschreiten. Dieser Strom kommt vonm Akku. Wenn Sie das RS232 Kabel am S4 angesteckt lassen, entlädt sich der Akku um 20% schneller. Wenn der S4 abschaltet, dann fällt die Spannung an der RS232 Schnittstelle auf Null.



Baud Rates

Die **BAUD RATE** einer Übertragung ist das Gegenteil die Zeit, die verwendet wird, um ein Bit zu senden. Die asynchrone serielle Übermittlung, die normalerweise verwendet wird, hat Extrabits um die Daten zu überprüfen: Ein **START** Bit, dann das Daten Bit, dann ein optionales Prüf Bit und anschl. ein oder mehr **STOP** Bits. Die **BAUD RATE** gibt die schnellst mögliche Übertragungsrate aber nicht die aktuelle Übertragungsgeschwindigkeit an. Wenn ein Computer seine maximale Übertragungsgeschwindigkeit erreicht hat, kann diese durch setzten einer höheren **BAUD RATE** nicht mehr gesteigert werden, die Abstände zwischen den Zeichen werden dadurch nur länger. Manche Systeme haben eine hohe **BAUD RATE** beim Senden, nicht aber beim Empfangen. Um Fehler zu vermeiden, werden die Signale **RTS** (**R**equest - **t**o - **S**end) und **DSR** (**D**ata - **S**et - **R**eady) verwendet. Diese Signale werden Handshake-Signale genannt. Leitungen die für serielle Übertragung verwendet werden, müssen unbedingt eine Handshake-Leitung haben, damit das Verlieren von Daten überwacht werden kann. Mit dem S4 ist eine Datenübertragung mit 115200 **BAUD** und einem Stoppbit möglich, wenn die **RTS** Leitung angeschlossen ist. Der S4 hat zwei aktive Handshake Signale in jede Richtung . **DTR** wird zusätzlich unterstützt und liefert immer „true", es gibt nur an, ob das Kabel angeschlossen ist. Das Signal auf dem Computer Ende, das Daten verlangt, ist nur **RTS** oder **DTR**. Sie können mit einem Meßgerät feststellen, welches auf einem hohen Level ist, und welches auf einem niedrigen. Es ist möglich, daß beide auf einem hohen Level liegen, wenn **DTR** auf ein positives Ende angeschlossen und **RTS** das Aktive Stop/GO Signal ist. Es ist möglich, alles ohne Handshaking zu betreiben, aber manche Computer schalten automatisch ab, wenn Sie kein korrektes Signal erhalten. Die Implementation ist wie folgt: Der S4 überprüft beim Einschalten das **CTS**. Wenn ein **CTS** Signal vorhanden ist, dann schreibt der S4 seine Daten auf die serielle Schnittstelle als auch auf das Display. Wenn kein **CTS** Signal beim Einschalten vorhanden ist, dann benutzt der S4 das Keypad und das **LCD** Display genauso wie die Schnittstelle für Eingaben. Wenn ein Kommando ankommt, dann gehen die nachfolgenden Meldungen auf das zugehörige Anzeigeinstrument.

ACHTUNG: *Beide Eingabegeräte sind bei dem Kommando **PROMPT** gleich geschalten. Wenn der S4 etwas in einem Modus bearbeitet, z.B. auf **RS232**, dann wird das Keypad ignoriert und umgekehrt. Etwas anderes würde wenig Sinn ergeben.*

Der S4 empfängt Kommandos über die serielle Schnittstelle in einem interaktivem Weg, z.B. werden Sie gebeten, für einen resultierenden Output zu warten. Normalerweise sind Computerprogramme dafür programmiert, um Kommandos anzunehmen, wenn sie eingegeben werden. Das verursacht normalerweise keine Probleme. In Bezug auf den Umstand, daß Kommandos von einem **BATCH** File kommen, welches nicht auf Ergebnisse wartet, müssen Sie sicherstellen, daß das **RTS** Handshake Signal auch wirklich vorhanden ist. Wenn Sie Files übertragen, dann vergessen Sie nicht, daß nicht alle Computer die Daten so schnell wie der S4 verarbeiten können. Um sicher zu gehen, senden Sie mit einer niedrigen **BAUD RATE** oder verbinden Sie die **RTS** Handshake Leitung. Sie können normalerweise eine schlechte Übertragung dadurch erkennen, wenn Sie das überspielte File ansehen. Es sind dann oft verkürzte Zeilen und falsche Zeichen zu sehen.

INTEL Format

Ein File im **INTEL** Format kann mit einem Text Editor angesehen werden. Jeder Eintrag ist eine einzelne Zeile von **ASCII** Zeichen, die jedes Byte als Hexadezimelcodepaar darstellen. Dies beinhaltet auch Carriage Return und Linefeed (**0D,0A**). Die Übertragung wird mit **ASCII** End - of - File (**1A**) beendet.

Erweiterter Adressen Eintrag

Position	Remarks
1	„**:**" ASCII Kolonnen Ende
2 - 3	„02" Byte Zähler
4 - 7	„0000" Adresse nicht verwendet
8,9	„02" Rekord Typen
10 - 13	Upper Segment Base Adresse. Top 16 Bits der 20 Bit Adressen. Auffallendste Byte am Anfang. 2 Bytes auf 4 Hex Zeichen. **USBA** Grundeinstellung ist 0 wenn nichts gesendet wird.
14,15	**CHECKSUM** Complement von allen Bytes außer eigenen Spalten, Linefeed und Return. 1 Byte entspricht 2 Hex Zeichen. Das macht die Zeile bis zu Null.
16,17	Carriage Return und Linefeed

Daten Rekord

Position	Remarks
1	„**:**“ ASCII Kolonnen Ende
2, 3	Datenbyte Zähler in HEX, max. „20“ (32 Bytes oder 64 ASCII Zeichen)
4 - 7	Niedrigeren 16 Bytes von 20 Bit Adressen. Auffallendste Byte am Anfang. 2 Bytes auf 4 Hex Zeichen. Diese müssen mit dem **USBA** von der 20 Bit Adresse verbunden werden.
8, 9	„01“ Daten Rekord.
10 - N	Daten Byte in Hexadezimal. Jedes Byte sind 2 Hex Zeichen.
N + 1, 2	**CHECKSUM** Complement von allen Bytes außer eigenen Spalten, Linefeed und Return.
N + 3, 4	Carriage Return und Linefeed

End of File Rekord

Position	Remarks
1	„**:**“ ASCII Kolonnen Ende
2, 3	„00“ Nummer von Bytes
4 - 7	2 Byte Übertragungs Adresse, normalerweise „0000“
8 - 9	„01“ Rekord Typ
10 - 11	**CHECKSUM** Complement von allen Bytes außer eigenen Spalten, Linefeed und Return, normalerweise „FF“
12 - 13	Carriage Return und Linefeed

Beispiel File im INTEL Format

:02000002100EC
:2000A0005BB66DDBB66CD8B060C183060D1B366DDBB66DDBB76FDFBF7EFDFBF6EDDBB66DD4
:2000C000DBB66DDAB56AD5AB56AD5BB66DDBB66DDBB66CD9B366CD9B366DDBB66DDBB66D34
:2000E000DBB76EDDBB76EDDBB66DDBB66DDBB66DDAB469D2A44992244992244992244993C5
:00000001FF

MOTOROLA S Format

Ein Motorola, Exorciser oder „S“ Format File kann mit einem Texteditor untersucht werden. Jeder Eintrag ist eine einzelne Zeile von **ASCII** Zeichen, die jedes Byte als Hexadezimelcodepaar darstellen. Jeder Rekord beginnt mit einem S. Dies beinhaltet auch Carriage Return und Linefeed (**0D,0A**). Die Übertragung wird mit **ASCII** End - of - File (**1A**) beendet.

S0 Header Rekord

Position	Remarks
1, 2	„S0“
3 - N	Kommentare
N + 1, 2	Carriage Return und Linefeed

S1 Daten mit 2 Byte Adresse

Position	Remarks
1, 2	„S1“
3, 4	Rekord Länge, 2 HEX Zeichen Nummer der Datenbytes plus 3 (2 Adressen & 1 Checksum).
5 - 8	2 Byte Adresse als 4 HEX Zeichen. Auffälliges erstes Byte.
9 - N	Daten Bytes in Hexadezimal, jedes Byte besteht aus 2 Hex Zeichen
N + 1, 2	Negierung der Checksumme der Bytes außer „S1“. Macht die Zeile bis „FF“
N + 3, 4	Carriage Return und Linefeed

S2 Daten mit 3 Byte Adresse

Position	Remarks
1, 2	„S2“
3, 4	Rekord Länge, Nummer der Datenbytes plus 4 (3 Adressen & 1 Checksum).
5 - 10	3 Byte Adresse als 6 HEX Zeichen. Auffälliges erstes Byte.
11 - N	Daten Bytes in Hexadezimal, jedes Byte besteht aus 2 Hex Zeichen
N + 1, 2	Negierung der Checksumme der Bytes außer „S2“. Macht die Zeile bis „FF“
N + 3, 4	Carriage Return und Linefeed

S8 End of File Rekord

Position	Remarks
1, 2	„S8“
3, 4	„04“ Rekord Länge.
5 - 10	3 Byte Start Adresse, normalerweise „000000“
11, 12	Negierung der Checksumme der Bytes außer „S8“. Macht die Zeile bis „FF“
13, 14	Carriage Return und Linefeed

S9 End of File Rekord

Position	Remarks
1, 2	„S9“
3, 4	„03“ Rekord Länge.
5 - 10	2 Byte Start Adresse.
11, 12	Negierung der Checksumme der Bytes außer „S9“. Macht die Zeile bis „FF“
13, 14	Carriage Return und Linefeed

Beispiel File im Motorola Format

```
S12300A05BB66DDBB66CD8B060C183060D1B366DDBB66DDBB76FDFBF7EFDFBF6EDBB66DD00
S12300C0DBB66DDAB56AD5AB56AD5BB66DDBB66DDBB66CD9B366CD9B366DDBB66DDBB66D30
S12300E0DBB76EDDBB76EDDBB66DDBB66DDBB66DDAB469D2A$499224499224499224499 3C1
S9030000FC
S2240100C0DBB66DDAD56AD56AB56AD5BB66DDBB66DDBB66CD9B366CD9B366DDBB66DDBB66D
D2E
S2240100E0DBB76EDDBB76EDDBB66DDBB66DDBB66DDAB469D2A4499224492244992244993BF
S804000000FB
```

Tektronix Format

Ein Tektronix Format File kann mit einem Texteditor untersucht werden. Jeder Eintrag ist eine einzelne Zeile von **ASCII** Zeichen, die jedes Byte als Hexadezimelcodepaar darstellen. Dies beinhaltet auch Carriage Return und Linefeed (**0D,0A**). Die Übertragung wird mit **ASCII** End - of - File (**1A**) beendet.

Tek Standart Daten Rekord

Position	Remarks
1	„/“ Slash Zeichen
2 - 5	4 HEX Zeichen zeigen die 2 Byte Ladeadresse, auffallendes erstes Byte.
6, 7	2 HEX Zeichen zeigen den Byte Count (Wie viele Daten Bytes keine Checksumme, % oder Adresse enthalten.
8, 9	2 HEX Zeichen zeigen die Checksumme der HEX Zeichen in der Load Adresse und im Byte Count an.
10 - N	Daten Byte, jedes Byte besteht aus 2 HEX Zeichen.
N + 1, 2	2 HEX Zeichen zeigen die Checksumme der HEX Zeichen in den Daten Bytes an.
N + 3, 4	Carriage Return und Linefeed

TEK End of File Rekord

Position	Remarks
1	„/“ Slash Zeichen
2 - 5	4 HEX Zeichen zeigen die 2 Byte Startadresse, auffallendes erstes Byte.
6 - 7	„00“ Byte Zähler
8, 9	2 HEX Zeichen zeigen die Checksumme der HEX Zeichen, außer „/“ an.
10, 11	Carriage Return und Linefeed

TEK Extended Data Rekord

Position	Remarks
1	„%“ Prozent Zeichen
2 - 3	2 HEX Zeichen zeigen die Zeichen Zähler im Rekord, außer „%“.
4	„6“ für Daten Rekord.
5, 6	Checksumme der HEX Zeichen außer „%“
7 - N	Zwischen 2 und 17 Zeichen. Die ersten HEX Zeichen spezifizieren wie viele HEX Zeichen auf die Load Adresse folgen, auffallendes erstes Byte.
N - M	Daten Bytes, jedes Byte besteht aus 2 HEX Zeichen.
M + 1, 2	Carriage Return und Linefeed

TEK Extended End of File Rekord

Position	Remarks
1	„%“ Prozent Zeichen
2 - 3	2 HEX Zeichen zeigen die Zeichen Zähler im Rekord, außer „%“.
4	„8“ für End of File Rekord.
5, 6	2 HEX Zeichen zeigen die Checksumme der HEX Zeichen, außer „%“ an.
7 - N	Zwischen 2 und 17 Zeichen. Die ersten HEX Zeichen spezifizieren wie viele HEX Zeichen auf die Start Adresse folgen, auffallendes erstes Byte.
N + 1, 2	Carriage Return & Linefeed

Beispiel eines Files im Tektronix Format

/00C0200EDBB66DDAB56AD5AB56AB56AD5BB66DDBB66DDBB66CD9B366CD9B366DDBB66DDBB6
6D58
/00E02010DBB76EDDBB76EDDBB66DDBB66DDBB66DDAB469D2A44992244992244992244993 0F
/01000001
%4C66E60100C0DBB66DDAB56AD546AB56AD5BB66DDBB66DDBB66CD9B366CD9B366DDBB66DD
BB66D
%C62560100E0DBB76EDDBB76EDDBB66DDBB66DDBB66DDAB469D2A449922449922449922244993
%0781010

ASCII Format

Jedes Daten Byte wird in 2 hexadezimale ASCII Zeichen übersetzt, es wird ein auffälliges Nybble Bite als erstes gesetzt. ASCII ist gleich wie INTEL, MOTOROLA und TEKHEX, außer, daß nichts außer den echten Daten gesendet wird, d.h. keine Adressen und keine Checksummen. Nach 32 Bytes oder 64 Zeichen werden ein Linefeed und ein Carriage Return gesendet (0D, 0A). Dies ermöglicht eine Übertragung zu einem Empfänger, bei einem seriellen Drucker oder Terminal. Am Ende einer Übertragung wird der End of File Charakter (1A) gesendet.
ASCII wird normalerweise als eine 8 Bit Übertragung gesendet und empfangen. Dies MSB wird low maskiert wenn empfangen wird, und low gesetzt, wenn gesendet wird.
Beim Empfangen fragt der S4 nach einer START und END Adresse, wohin der Block gespeichert werden soll.

Achtung: *INTEL, MOTOROLA und TEKHEX Übertragungen können vom Empfänger überprüft werden, wenn diese im ASCII Format ankommen. Dann können Sie Delimiters, Adressen und Cheksummen überprüfen.*

Beispiel File im ASCII Format

3B	77	EE	DD	BB	77	EE	DD	BB	77	EE	DD	BA	74	E9	D2
A5	4B	96	2D	5A	B4	69	D2	A5	4B	96	2C	58	B1	63	C6
8D	1B	36	6C	D8	B1	63	C6	8D	1B	37	6F	DE	BD	7B	F6
ED	DB	B7	6F	DE	BD	7B	F6	ED	DA	B5	6B	D6	AD	5B	B6
6D	DA	B5	6B	D6	AD	5B	B6	6C	D9	B2	64	C9	92	24	49
93	26	4D	9B	36	6D	DB	B7	6E	DC	B8	71	E3	C7	8E	1D
A5	4B	96	2D	5A	B4	69	D2	A5	4B	96	2C	58	B1	63	C6
ED	DB	B7	6F	DE	BD	7B	F6	ED	DA	B5	6B	D6	AD	5B	B6
6D	DA	B5	6B	D6	AD	5B	B6	6C	D9	B2	64	C9	92	24	49
8D	1B	36	6C	D8	B1	63	C6	8D	1B	37	6F	DE	BD	7B	F6
3B	77	EE	DD	BB	77	EE	DD	BB	77	EE	DD	BA	74	E9	D2
93	26	4D	9B	36	6D	DB	B7	6E	DC	B8	71	E3	C7	8E	1D
A5	4B	96	2D	5A	B4	69	D2	A5	4B	96	2C	58	B1	63	C6

BINARY Format

Das File wird als eine Zusammensetung von Bytes gesendet. Die Daten werden nicht überprüft. Der S4 fragt Sie um eine **START** und eine **END** Adresse beim Empfang. Es wird kein **ASCII** End of File Charakter mehr gesendet, da dies beim **BINARY FORMAT** als Eingangsdaten gedeutet würden. Das Senden von Binär Dateien mit dem S4 ist möglich, kann aber beim Empfänger zu Schwierigkeiten führen. Der Computer muß einen genau vordeffinierten Block mit bestimmter Größe auf einmal, an eine vorbestimmte Stelle speichern. Wenn ein Block im Binär Format empfangen wurde, hat der S4 keine Ahnung ob der Computer schon alles gesendet hat. Sie müssen deshalb mit ESCape die Übertragung abbrechen.

Achtung: Computer können keine Binären Daten über die serielle Schnittstelle senden, außer Sie verwenden einen Zusatz, wie z.B. bei Copy /B, bei PIP [O]. Anderenfalls würde die Übertragung abgebrochen, wenn ein End of File (1A) Zeichen gefunden wird.

BINÄRE Übertragungen sollten mit anderen File Formaten überprüft werden, da jedes Zeichen, das im Speicher vom S4 erscheint, überprüft wird, ob es wirklich angekommen ist. Alles inklusive Carriage Returns, Linefeeds etc..